

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ЕЁ ПРИМЕНЕНИЕ

Новое в жизни, науке, технике

Подписная научно-популярная серия

Издается ежемесячно с 1988 г.

Экспертные системы

1990

10

Новое
в жизни,
науке,
технике

ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И ЕЁ ПРИМЕНЕНИЕ

Подписная
научно-
популярная
серия

10/1990

ЭКСПЕРТНЫЕ СИСТЕМЫ

Издается
ежемесячно
с 1988 г.

В номере:



Издательство
"Знание"
Москва
1990

ББК 32.97
Э 41



ГОЛУБЕВА Людмила Николаевна — кандидат философских наук, старший преподаватель, специализируется в области экспертный систем.

ПОКАТАЕВА Елена Николаевна — научный сотрудник МЭИ.

ЕЗДАКОВ Андрей — инженер, программист.

ЧАСТИКОВ Аркадий Петрович — кандидат технических наук, доцент. Научные интересы — эволюция СУБД и экспертных систем.

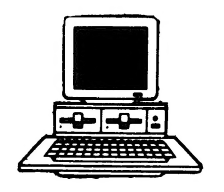
ДЕДКОВА Татьяна Геннадьевна — инженер, программист.

МОРОЗОВ Виктор Александрович — кандидат технических наук, специалист в области программного обеспечения САПР.

КЛЮЧКО Владимир Игнатьевич — доктор технических наук, специалист в области информационных систем.

РЕДАКТОР Б.М.ВАСИЛЬЕВ

ISBN 5-07-000529-4



Хотя эвристический подход играет большую роль в построении систем распознавания образов, не много может быть сказано относительно общих принципов синтеза, так как решение каждой конкретной задачи требует использования специфических приемов разработки системы. Это означает, следовательно, что структура и качество эвристической системы в значительной степени определяются одаренностью и опытом разработчиков.

Дж.Ту, Р.Гонсалес

Л.Н.Голубева

Экспертные системы — осторожный оптимизм

Из истории экспертных систем

Первые ЭС — медицинская MYCIN и DENDRAL для приложений в химии появились в середине 70-х годов в рамках исследовательских программ по искусственному интеллекту (ИИ).

Один из участников группы, работающей над DENDRAL, Э.Фейгенбаум считал, что главная причина неудач ИИ 60-х годов заключалась в том, что в машину закладывались произвольные знания, об эффективности которых мы ничего не знаем. По его мнению, недостаток исчезнет, если сузить задачу, т.е. использовать в машине лишь знания эксперта — высококвалифицированного специалиста в узкой предметной области. Эта гипотеза была проверена на конкретном материале — медицинском знании. Обращение к медицине — не случайность. С древних времен врачи ставили диагноз, исходя из наблюдений больного и своего опыта. Результаты наблюдений обобщались в учении о нозологических формах болезней, в котором врач находил описание эмпирических ассоциаций опытных специалистов по распознаванию той или иной болезни. Основной инструмент врача — фиксация симптомов и вывод — построение цепочки силлогизмов. Так вот, большие посылки силлогизмов напоминают продукции — формализмы представлений знаний, открытые А.Ньюэллом и Г.Саймоном в 1972 году. Действия же врача-диагноста напоминают своеобразный алгоритм (рис.1) и могут быть вполне удовлетворительно описаны в терминах "формальной продукционной системы", предложенной Постом и позже реализованной в архитектуре продукционных ЭС.

Гипотеза Э.Фейгенбаума блестяще подтвердилась. Система, базирующаяся на экспертном знании, приобрела не виданные для интеллектуальных систем того времени свойства. Она умела "встраиваться" в среду непрограммируемого пользователя, приспосабливаясь к его запросам. Благодаря этому ЭС стали новой информационной технологией, предназначенной для автоматизации в тех областях, где есть опыт экспертов, базирующийся на понимании структуры и назначении конкретных знаний.

Уже первые ЭС оказались полезными. Медицинская система MYCIN успешно вписалась в клиническую практику, помогая в выборе лекарств больным с бактеремией, менингитом, циститом. Давно было замечено, что в обычных условиях врач широкого профиля-терапевт обычно принимает неэффективные решения, назначая лекарства с широким спектром действия. ЭС же дает более точное предписание лечения. То же самое можно сказать о системе DENDRAL, работа над которой началась еще в 1965 году! Она помогает химику-органику определять химическую структуру синтезированных химических веществ при наличии данных масс-спектрометрии. Если же проводить эти работы вручную, то неизбежны ошибки, ибо надо перебрать очень много структур. Благодаря чему ЭС достигают такого эффекта?

Идеологию ЭС можно выразить формулой: Знание плюс вывод =

Система. ЭС предполагает взаимодействие блоков. Главные из них — база знаний и механизм вывода. Кроме того, имеется и развитый интерфейс с пользователем и модуль пополнения знаний (рис.2). Так, работа ЭС MYCIN начинается с вопросов, обращенных к пользователю. В ходе диалога выявляются данные о пациенте и происходит обсуждение результатов анализов. Выдвигается предварительная гипотеза. Система просматривает базу знаний и выбирает те, правая часть которых совпадает с предположением. Так как проблемная область содержит неоднозначные предположения и врач до конца не уверен в диагнозе, то для активированных правил в базе знаний вычисляются коэффициенты уверенности по специальным формулам. При необходимости система запрашивает дополнительную информацию (в тех случаях, когда коэффициент уверенности для какого-либо факта не может быть определен с помощью коэффициентов уверенности исходных фактов). Затем выдается рекомендация.

Названные системы оказались "родоначальниками" многих дочерних ЭС. Разработчики вторглись в промышленную сферу. Ремонт, техническая диагностика, обслуживание промышленных процессов, геологоразведка — самые популярные для разработчиков ЭС предметные области. Наряду с этим ЭС продолжают осваивать сферу медицины и образования. Процесс появления новых ЭС можно уподобить дереву, фазы роста и конечные результаты которого определяют открытия в области новых архитекур. Последних значительно меньше, чем самих ЭС. Это связано со сходством стратегий решения проблем в разных предметных областях и однотипностью экспертного знания. К примеру, архитектуры, реализованные в геологической ЭС PROSPECTOR и ЭС MYCIN, базирующиеся на продукционных системах со встроенным механизмом комбинирования "уверенностей", были воспроизведены в ЭС, предназначенных для сельского хозяйства, анализа механических конструкций, имитационного моделирования. То же самое происходит и с другими архитектурами. Некоторые из них ориентированы на "глубинное знание" — причинные модели, категории, абстракции. Так, в медицинской системе INTERNIST используются знания о взаимосвязях болезней, вследствии чего система "может поставить" дифференцированный диагноз. На распространение ЭС влияет не только наличие "подходящих" архитектур, но и нетехнические факторы: доступность экспертного знания, достоверность экспертизы, необходимость автоматизации, неэффективность традиционных способов трансляции высококачественных специальных знаний.

Экспертные системы как объект оценки

В начале 80-х годов ЭС появляются на рынке средств индустрии информации и сразу завоевывают признание. Фирмы и организации видели в ЭС источник конкурентоспособности, повышения качества деловой стратегии и культуры работы. Однако отношение к ЭС со стороны профессионалов-смежников достаточно прохладное. Так, философы размышляют о последствиях внедрения ЭС, об изменениях в этой связи статуса знания и "знающего человека". Поставлен вопрос даже об ограниченности человеческого знания и понимания как о факторе, который сможет затормозить развитие ЭС как новой информационной технологии [1]. Но самое прохладное отношение к ЭС у специалистов по ИИ, которые считают инженерию знаний "иждивенкой" интеллектуального капитала ИИ 70-х годов.

Разноречивость оценок — показатель многоплановости феномена ЭС. ЭС — это система симбиотического типа, как, к примеру, биогеоценоз. Для конкретной ЭС существует ее "местообитание" — среда, населенная людьми, основная черта которой состоит в потребности зафиксировать

уровень высококачественных знаний и усвоения их. Вот почему ЭС обнаружили в себе свойства "коммерческого продукта", носителя товара — знания. ЭС незаменимы в тех ситуациях, когда пользователь должен принимать решение, выходящее за рамки его компетентности. Пример: ремонт космической станции на орбите силами космонавтов, которые не знакомы со всеми бортсистемами в деталях. Другим фактором, вынуждающим использовать ЭС, выступает наличие экстремальных ситуаций, когда пользователь должен получить квалифицированную помощь в более сжатые сроки, чем это делается людьми (пример: управление производственными и технологическими процессами в реальном масштабе времени). Обозначенные ситуации — показатель того, как в современном обществе изменилось отношение к знанию. Знание рассматривается как семантический ресурс практической деятельности. Не случайно критерий оценки ЭС носит технократический характер: ЭС должна давать правильные заключения в 90-95% случаев, иметь "дружелюбный интерфейс" с пользователем и быстрый отклик на запрос.

А сейчас рассмотрим возражения специалистов по ИИ.

Прежде всего в литературе отмечается неспособность ЭС в полной мере симитировать деятельность специалистов-экспертов. Харт называет ЭС "тупыми слугами" человека, а Р.Шенк указывает на их неумение "нарушать" правила, как это делает человек-эксперт. Действительно, современные ЭС имеют "минимальное самосознание": они не владеют здравым смыслом, не рефлексируют по поводу своих действий, не открывают новое научное знание, не могут вобрать в себя ценностные ориентации человека. Для них невозможны такие привычные для человека роли в творческой коммуникации: критик, энциклопедист, генератор идей. Далее, ЭС представляют собой не что иное, как "решатель задач" поискового типа, и это грубая абстракция деятельности человека. ЭС не могут непосредственно "войти" в большой круг циркуляции знаний в системе.

Указанные недостатки нельзя разрешить простым указанием на них. Разработчики ЭС еще не располагают целостной концептуальной картиной интеллектуальной деятельности человека. Пока еще разработаны ее отдельные фрагменты, содержащие разнородное знание. Среди них можно обозначить:

1. Работы в области "обучающейся машины" (ИИ, кибернетика).

2. Выработка инструментария для синтеза разнородных знаний с ориентацией на компьютер.

3. Формирование антропологии "интеллектуальных систем" (изучение способов построения "когнитивных окон для человека" — термин принадлежит Д.Мичи).

Немного о первом фрагменте: в литературе есть ссылки на ряд "обучающихся" ЭС, самой популярной из них является EURISCO, которая тестировалась на задачах в военном деле и разработке логических элементов СБИС. Однако уникальность систем, различие в их архитектурах, идеологиях, языках представления знаний свидетельствуют о том, что идея "обучения" ЭС находится еще в зачаточной стадии. Лишь на специальном материале достигается успех этих программ. Это показатель того, что ИИ даже в "браке" с кибернетикой не может разрешить сам проблему обучения с тем, чтобы "передать" свой опыт разработчикам ЭС. Гарантией разрешения проблемы может выступить только совместное решение без навязывания иным дисциплинам каких-то узких концептуальных схем. Исходной же точкой отсчета должен стать базовый принцип новых информационных технологий "Технологическое отношение к знанию", ибо вряд ли машина будущего сможет одна, без человека загружать и пополнять свою базу знаний.

В этом плане заслуживает внимания идея С.Осуга о "разрабатывающей форме представления знаний" [2] — машинной поддержке интел-

лектуальной деятельности пользователя в порождении новых объектов и отыскании новых знаний с помощью быстрой реализации цикла построения моделей, их анализа, оценки. Такие системы в отличие от ЭС, ориентированных на "поверхностные", эмпирические знания, не относящиеся к структуре объектов, смогут обрабатывать знание, повышая его уровень системности, интегральности. К примеру, в машину будут "закладываться" частичные, фрагментарные знания, а в итоге сформируется синтетическое знание о плохоформализуемых, диффузных объектах. Однако такая форма представления знаний потребует иной техники описания проблем и знаний, а также архитектуры системы. Объектом описания станет не только личностное знание, но и парадигмальное — достояние научного сообщества в виде дисциплинарной матрицы (термин принадлежит Куну), а также "внесубъектное" знание (принципы, теории, методология). Поэтому место инженера знаний займет метаинтерпретатор-когнитолог, умеющий работать с различными видами знаний, вооруженный современной техникой метоязыка для синтеза.

Замечание о третьем фрагменте — антропологии ЭС. Проблема человека за дисплеем — ключевая для всех информационных технологий. И хотя ЭС включает в себя человекоориентируемые элементы: доступ к "профессиональной культуре", "доверительное" общение с "непрограммируемым" пользователем, тем не менее нельзя сказать, что проблема решена. Современные ЭС не имеют "когнитивных окон" для человека, ибо действия системы зачастую носят скрытый, необъяснимый характер. Так, объяснение в продукционных ЭС базируется на демонстрации "трассы" вывода рекомендаций, т.е. показе "машинных" рассуждений. Поэтому нет эпистемологических, понятных для человека рамок показа. Это дополнительный аргумент в пользу систем с "разрабатывающей формой представления знаний".

Типы современных ЭС

Первый тип — это так называемые ЭС первого поколения. Их отличительная особенность состоит в том, что системы предназначаются для решения хорошо структурированных задач, требующих сравнительно небольшого объема эмпирических знаний. Обычно это либо классификационные задачи, либо задачи выбора из имеющегося набора вариантов. Назовем основные черты ЭС первого поколения.

1. Узкая область экспертизы.
2. Развитая система общения с "непрограммируемым" пользователем. Последний — источник информации о конкретной ситуации, и поэтому система должна правильно оценивать содержание естественно-язычных сообщений.
3. Ограниченные возможности в модификации ЭС в процессе эксплуатации. Как правило, необходим посредник — инженер знаний, который отвечает за форму представления экспертных знаний в компьютере.
4. Основной вид используемых знаний — личностное знание, отражающее профессиональный опыт эксперта.

Успех в разработке ЭС первого поколения во многом зависит от профессиональной компетенции инженера знаний. Это и новая профессия, которая требует гармоничного сочетания двух типов мышления: гуманитарного и "программистского". Гуманитарная составляющая необходима для концептуальной оценки предметной области, проблем. Кроме того, инженер знаний должен владеть навыками практического психолога, для того чтобы извлечь у эксперта информацию о стратегиях решения задач. "Программистская" составляющая необходима для того, чтобы оценивать возможности инструментальных средств инженерии знаний (языков программирования, систем программирования и др.). Таких специалистов мало и это сдерживает распространение ЭС данного типа. По замечанию Д.Уотермена, новые ва-

кансии в фирмах "появляются гораздо быстрее, чем наши университеты могут подготавливать докторов философии по искусственному интеллекту" [3].

Существует и другой тип ЭС, так называемые "оболочки ЭС". Это метасистемы, предназначенные для приобретения знаний из внешней среды. Основой оболочки выступает архитектура — "готовый" механизм вывода и "пустая" база знаний. Приобретение знаний поддерживают сервисные программы — редактор базы знаний, средства отладки, трассировки, предназначенные для инженера знаний. Основная трудность в их использовании состоит в настройке на конкретную предметную область. Оболочка диктует разработчику язык описания фактов, формализм представления знаний и стратегию вывода. Естественно, что не все задачи укладываются в эту формальную рамку.

Третий тип — "гибридные" ЭС и ЭС в реальном масштабе времени. Их главный признак — использование ЭС как методологического средства для информационного дизайна, цель которого состоит в создании больших систем обработки информации. По сути дела, это не экспертные системы в их каноническом виде. ЭС лишь элемент, "интеллектуальный интерфейс", позволяющий пользователю выходить на объективированное знание, заключенное в базах данных, пакетах прикладных программ. Примером может служить ЭС FLUX EXPERT, предназначенная для решения уравнений в частных производных с использованием САПР [4]. Экологическая ниша для таких систем — необходимость выполнения задач с использованием методов системного анализа, исследования операций, математической статистики, прикладной математики, обработки информации.

Четвертый тип ЭС — сети ЭС. Их отличительная особенность состоит в "связывании" нескольких ЭС, когда решения одной из них выступают исходным сырьем для действий другой. Они эффективны для задач с интегрированной структурой решений в условиях распределенной обработки информации. В переводной литературе есть ссылка на сеть — EXMAT, предназначенную для химической идентификации многокомпонентных материалов (сеть сопряжена с лабораторным оборудованием).

Каждый тип ЭС из приведенных выше не разделен перегородками от предыдущего, более того, наблюдается тенденция к разнообразию видов используемых знаний. Преимущество ЭС заключается в использовании "неявного знания" — спутника познавательной деятельности человека. Это создает расширенные возможности для активизации информационных ресурсов общества, ибо "компьютерное отчуждение" неявных знаний устраняет барьеры между человеком и совокупным фондом накопленных человечеством профессиональных знаний.

Литература

1. Экспертные системы. Принципы работы и примеры/Пер.с англ. А.Брукинг, П.Джонс, Ф.Кокс и др. Под ред. Р.Форсайта. — М.: Радио и связь, 1987. — 221 с.

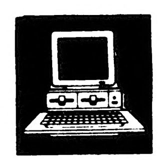
2. Осуга С. Обработка знаний/Пер. с япон. — М.: Мир, 1989. — 235 с.

3. Уотермен Д. Руководство по экспертным системам. — М.: Мир, 1989. — 184 с.

4. Купон Ж.-Л., Сабоннадьер Ж.-К. САПР в электротехнике/Пер, с франц. — М.: Мир, 1988. — С.154-168.

5. Искусственный интеллект: применение в химии/Пер.с англ. Д.Смит, Ч.Риз, Дж.Стюарт и др. Под ред. Т.Пирса. — М.: Мир, 1988. — С.403-416.

6. Хьюитт К. Открытые системы//Реальность и прогнозы искусственного интеллекта: Сб.ст.. — М.: Мир, 1987. — С.86 (Хьюитт).



Именно в свете идей и результатов кибернетики стало очевидным общенаучное, гносеологическое значение формализации как метода, который в принципе применим (в той или иной форме) ко всякой области конкретных или абстрактных объектов познания.

Б.В.Бирюков

Е.Н.Покатаева

Черное золото 90-х годов, Или как видеть мир глазами другого человека

Искусственный интеллект имеет богатую историю развития, в которой можно выделить значимые этапы:

- имитация человеческого разума с помощью нейронных сетей (система PERCEPTRON) — 50-е годы;

- увлечение эвристическими алгоритмами, создание системы GPS, объявленной универсальной по отношению к всевозможным предметным областям — 60-е годы;

- осознание невозможности построения универсальной модели предметной области, эффективной для реальных ситуаций;

- переключение внимания на очень узкие предметные области и интенсивные разработки эффективных способов представления и манипулирования знаниями, которые привели к созданию реально работающих экспертных систем (MYCIN, PROSPECTOR, PUFF), делающих правдоподобные выводы в условиях неточных, ненадежных исходных данных.

Современный этап, по выражению Р.Форсайта, является "очередным раундом необузданного оптимизма, охватывающего не только замкнутую в себе область искусственного интеллекта, но и всю совокупность людей, занятых информатикой, вычислительной техникой и обработкой данных" [1]. Основанием для этого необузданного оптимизма является уверенность в том, что именно знания, их обширность и качество, являются тем магическим ингридиентом, который определяет успех функционирования экспертной системы. "Знания (в машинно-приемлемой форме) могут стать "черным золотом" 90-х годов... Те люди, которые хотят открыть самую мощную скважину, будут в числе самых первых, кто займется массовым производством знаний с помощью компьютера" [1]. Фаза "массового производства знаний" потребовала появления особой отрасли человеческой деятельности — инженерии знаний и специалистов в этой области — инженеров по знаниям.

Инженерия знаний — очень молодая область и как научное направление находится на этапе становления. Поэтому остро ощущается нехватка соответствующих специалистов — инженеров по знаниям, которые должны владеть методами представления и использования знаний, методами программирования, обладать высокой эрудицией, чтобы эффективно трудиться в различных предметных областях, а также отчетливо представлять себе, какие именно знания, откуда, каким образом приобретать и каким способом преобразовывать в машинно-ориентированную форму. Таких специалистов во всем мире не так уж много.

Инженерия знаний решает две главные задачи: приобретение и накопление знаний; обработка и использование их. Этап приобретения и накопления знаний, заканчивающийся формированием базы знаний (БЗ) экспертной системы, является чрезвычайно важным этапом в технологии построения ЭС и одновременно "узким местом" современных ЭС.

Важность данного этапа определяется тем, что ЭС в сравнении с человеком, который непрерывно приобретает знания из своего собственного опыта в течение всей жизни, способна

лишь систематизировать, проверять на непротиворечивость и надежность те знания, которые получены инженером по знаниям из бесед с экспертами-специалистами, а также из руководств и справочников. Следовательно, качество знаний ЭС определяется уровнем квалификации как эксперта, так и инженера по знаниям.

Данный этап технологии разработки пока критический для современных ЭС, поскольку не существует разработанной методологии, применение которой к конкретной ЭС позволит получить по крайней мере удовлетворительный ответ на вопрос: "откуда, что и как приобретать?"

Источники знаний

Источниками знаний для конкретных ЭС могут быть учебники, справочники, материалы конкретных исследований в проблемной области и т.п. Сами разработчики могут иметь теоретические знания и практический опыт в данной области. Но классическим источником знаний является эксперт-профессионал в данной предметной области. Таким образом приобретение знаний требует учета человеческого фактора. Для успешного решения этой проблемы необходимы совместные усилия математиков, программистов, психологов.

Имеется опыт работы психологов, исследовавших вопрос: что такое эксперт? Имеет ли он качественные отличия в чем-либо от не-эксперта? Любой ли специалист может быть экспертом?

На ранних этапах исследований предполагалось, что любой новичок — это тот же эксперт, знания которого не наполнены содержанием. Если же дать ему недостающие знания, из новичка автоматически получится эксперт, причем чем больше знаний, тем выше квалификация эксперта. Здесь можно провести аналогию с различными видами реализаций экспертных систем. Кроме готовых ЭС с заполненной БЗ, разрабатывается большое количество систем с незаполненными, "пустыми" БЗ — так называемые "оболочки ЭС". Предполагается, что заполнение БЗ конкретными знаниями (например, продукционными правилами) с помощью небольшого встроенного редактора базы знаний позволяет очень быстро получать полноценные ЭС, работающие на уровне эксперта в выбранной предметной области.

Дальнейшее изучение поведения новичка и эксперта привело к выводу, что оба типа специалистов используют качественно различный подход к решению задач в своей области. Главным моментом здесь отличают способность человека фиксировать в своем сознании промежуточные вспомогательные точки пути решения задачи и представлять их вербальное описание.

Накопленный опыт создания ЭС свидетельствует, что эксперты имеют тенденцию формулировать свои заключения и то, как они пришли к ним, в общих выражениях, которые слишком широки и расплывчаты для их анализа и обработки на ЭВМ. Высококвалифицированный эксперт быстро делает сложные выводы, не утруждая себя тщательным анализом каждого шага процесса рассуждения. Кроме того, очень многие знания эксперт использует автоматически, интуитивно, не подозревая о том, что эти знания служат результатом сложного логического процесса, основанного на данных его богатой практической деятельности и огромных массивов информации, хранящихся в человеческой памяти. Эксперт может назвать свои заключения предчувствием, догадкой, озарением, и будет при этом совершенно искренен и, главное, прав. Сложность человеческой психики и мыслительной деятельности делает невозможным формирование таких вербальных сообщений, которые, во-первых, были бы доступны обработке на ЭВМ по степени конкретизации, а во-вторых, отражали бы глубинные зависимости объектов предметной области на уровне высококвалифицированного эксперта.

Более того, опыт работы с различ-

ными экспертами привел Д.Уотермана к выводу, который он сформулировал в [2] как парадокс экспертизы или парадокс инженерии знаний: "Чем более компетентными становятся эксперты, тем менее способны они описать те знания, которые используют для решения задач!" Причем оказалось, что когда эксперты пытаются объяснить, как они пришли к заключению, они зачастую выстраивают правдоподобные линии рассуждения, которые, однако, мало похожи на их действительное поведение при решении задач. Эта особенность имеет очень важные для процесса приобретения знаний следствия, которые Уотермен выразил с помощью советов инженерам по знаниям:

1. Не будьте своим собственным экспертом.

2. Не верьте всему, что говорят эксперты. Каждое экспертное правило будет законным не потому, что эксперт ручается за его правильность, а потому, что эксперт продемонстрирует убедительное число раз применение этого правила при решении задач.

Таким образом, основной источник знаний для ЭС — эксперт — источник крайне ненадежный, и пока нигде и никем в мире не предложено удовлетворительной общей методики, позволяющей приобретать достаточный объем достоверных знаний. Однако рынок ЭС расширяется, Д.Уотермен в [2] приводит каталог многих десятков ЭС и инструментальных средств для их создания почти в двух десятках сферах человеческой деятельности (медицина, геология, физика, юриспруденция, космическая техника и т.д.). Проблема приобретения знаний все же решается.

Способы приобретения знаний

И все-таки мы можем увидеть то, что стоит за словами.

Дж.Келли

Убедившись в том, что наличие человеческого фактора в решении проблемы приобретения знаний делает эту проблему вряд ли разрешимой в ближайшее время, логично обратиться к идее автоматизации процесса приобретения знаний, устраняющей присущую человеку нечеткость и непоследовательность рассуждений.

На настоящий момент нет готовых систем, позволяющих исключить человека из цепочки, причастной к формированию БЗ. Однако имеются теоретические исследования и исследовательские разработки, наличие которых позволяет выделить три класса способов приобретения знаний:

1. Традиционный диалог эксперта с инженером по знаниям, в котором все знания предоставляются экспертом.

2. Автоматическая генерация знаний, которая позволяет часть правил получать автоматически.

3. Построение индивидуальной модели исследования предметной области конкретным экспертом, позволяющей организовать целенаправленный процесс исследования этой области на основе индивидуальных представлений данного эксперта.

Традиционный диалог с экспертом

Традиционный способ состоял в том, чтобы запереть на несколько месяцев высокооплачиваемого домашнего эксперта (или пару) вместе с высокооплачиваемым специалистом по технологии знаний, в течение которых они эффективно обговорят закодированный вариант того, что знает специалист.

Р.Форсайт

Этот процесс состоит из продолжительной серии напряженных систематических интервью, обычно продолжающихся в течение многих месяцев. Во время этих бесед инженер по знаниям предлагает эксперту реальные задачи или близкого к ним типа. А взамен получает парадокс инженерии знаний и его два следст-

вия, а также нежелание эксперта работать в этом качестве, поскольку:

- у него есть своя работа, которой необходимо заниматься;

- он опасается, что может потерять работу, если его — высококлассного, высокоценимого и высокооплачиваемого специалиста — заменит компьютер;

- он чувствует угрозу вторжения компьютера в свою приватную область;

- он не хочет, чтобы на его "профессиональную кухню" оценивающе смотрел другой эксперт, пусть даже более низкого уровня.

Это значит, что инженер по знаниям должен не только удовлетворять тому списку требований, который мы привели выше, но и обладать дополнительными качествами, как то: коммуникабельность, умение завоевывать доверие и располагать к себе людей. Кроме того, этап непосредственного общения с экспертом обязательно предшествует этапу формирования рабочих гипотез, подготовки тестовых задач для проверки этих гипотез.

Идея о том, что заключения экспертов имеют характер интуитивной догадки, привела к созданию модели интуитивного озарения (рис.1) и использованию этой модели в диалоге с экспертом.

Модель интуиции предполагает, что знания эксперта имеют ряд образцов-указателей, помогающих выявлять типичные ситуации и обращаться к необходимым участкам знаний. На рис.1 показана ситуация распознавания факта, принадлежащего текущей области компетентности эксперта. Когда эксперты сталкиваются с новой ситуацией, они пытаются применить общие принципы и реализовывают дедуктивные шаги, чтобы установить причинно-следственные связи данной ситуации с имеющимися шаблонами, Неплохо, если бы, замечает Д.Уотермен, "экспертные системы осуществляли вывод из фундаментальных принципов, когда они сталкиваются с незнакомыми проблемами, но как это сделать реально, мы еще не знаем". Пока что эти стратегии предлагают один из способов взаимодействия с экспертом, заключающийся в предъявлении эксперту новых или неожиданных для него ситуаций и наблюдение при этом за его поведением.

Существуют другие способы организации диалога с экспертом. В книге [3], написанной крупными японскими специалистами в области инженерии знаний, приемы техники извлечения знаний из экспертов делятся на шесть основных классов.

1. Опрос с наводящими вопросами — метод, наиболее близкий к традиционному диалогу. Инициатива в этом опросе принадлежит целиком одному из партнеров, который предварительно готовит тему диалога, выделяет список необходимых вопросов.

2. Структурированный опрос, в ходе которого инженер по знаниям строит модель проблемной ситуации, уточняет понятия предметной области, формирует зависимости между понятиями.

3. Самонаблюдение, в ходе которого эксперт размышляет вслух и говорит все, что приходит в голову по поводу заданной ситуации. Этот своеобразный "поток сознания" позволяет получить общую информацию о типах используемых знаний, стратегиях решений, применимости тех или иных подходов.

4. Самоотчет — разновидность рассуждения вслух, когда эксперт

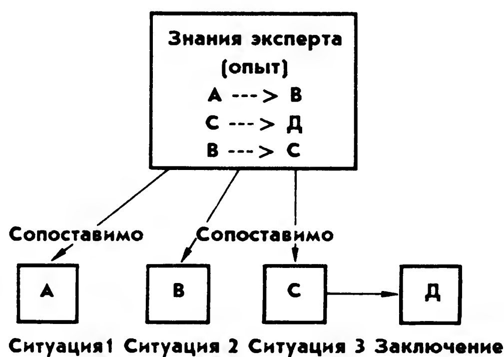


Рис. 1

одновременно с описанием хода рассуждений дает самому себе отчет, почему он предпринял те или иные шаги, отклонил те или иные гипотезы.

5. Консультация, представляющая собой взаимодействие специалиста с инженером по знаниям с целью получения рекомендаций о том, каким образом целесообразно решать те или иные проблемы.

6. Критический обзор — наиболее удобная форма приобретения метазнаний, т.е. знаний о том, как применять проблемные знания.

Каждый из указанных приемов организации диалога заканчивается этапом анализа протоколов общения с экспертом.

Как мы видим, использование человека-эксперта в качестве единственного источника знаний для ЭС сопряжено с длительными сеансами общения эксперта с инженером по знаниям. Причем участие эксперта необходимо не только на этапе первичного выявления знаний, но в процессе всех последующих стадий разработки ЭС. Можно привести такую классификацию методов приобретения знаний в соответствии с различными этапами разработки системы:

1. Наблюдение на рабочем месте (наблюдать за тем, как эксперт решает реальные задачи).

2. Обсуждение задач (выявление необходимых типов данных знаний, процедур).

3. Описание задач (формирование задач-прототипов для всех возможных категорий задач).

4. Анализ задачи (наблюдение за решением экспертом реальных задач различных категорий).

5. Доводка системы (формирование реальных задач на использование выявленных правил).

6. Оценивание системы (критическая оценка работы системы).

7. Проверка системы (проверка работоспособности системы на примерах, предоставленных другими независимыми экспертами).

Кроме того, по расстановке акцентов в процессе работы с экспертом имеются три подхода к приобретению знаний:

1. Создание долговременной рабочей группы из экспертов и инженеров по знаниям, работающих совместно на протяжении всего периода создания ЭС. Таким образом разрабатывались известные ЭС PROSPECTOR, INTERNIST и др.

2. Оперативное создание прототипа на основе первичных знаний о проблемной области, что призвано в основном вызвать энтузиазм у экспертов для продолжения совместных работ.

3. Тщательный анализ знаний, который предшествует началу реализации системы, который позволяет при необходимости не поддерживать связи с экспертами на этапе проектирования и практической реализации системы.

Можно ли каким-либо образом оптимизировать время, необходимое для диалога с экспертом, или улучшить качество приобретаемых знаний? Имеются отдельные экзотические предложения. Например, в [5] предлагается повысить эффективность извлечения глубинных знаний эксперта за счет погружения человека в особые психические состояния, в которых "происходит глубокий ассоциативный поиск,диалог сознания и подсознания человека". Это особое психическое состояние активизирует определенные понятийные структуры за счет механизма неосознанного ввода корректирующих информационных символов, эффекта наложения так называемых мотивационных приоритетов различных понятийных структур.

Данный метод — интересная исследовательская разработка, которая не может быть предложена в качестве инструмента всей армии разработчиков ЭС.

Таким образом, приобретение знаний в процессе диалога с экспертом — процесс трудоемкий, с трудно прогнозируемым результатом, требующий больших временных затрат и психологических нагрузок для всех участников процесса. Поэтому

совершенно естественно, что по мере развития и практического применения техники обработки знаний повышался интерес к таким вопросам, как обучение и приобретение знаний с помощью компьютеров. Неизбежно появилось стремление к созданию систем с функциями приобретения, хранения и применения знаний.

Автоматическое приобретение знаний

Под обучением машины я понимаю любое улучшение работы вычислительной системы, являющееся результатом накопления опыта.

Р.Форсайт

Большинство реальных систем, способных к обучению, разработанных к настоящему моменту, ориентировано на анализ механизма приобретения информации человеком и его воспроизведение в виде компьютерной программы. Человеку свойственно эвристическое обучение. Программа EURISKO, появившаяся в 1932 г., автоматически улучшает и расширяет свой запас эвристических правил. Эта система выигрывала 3 года подряд в учебной военной игре, несмотря на то что правила игры каждый раз менялись. А также произвела переворот в области технологии (сверхбольших интегральных схем, изобретя трехмерный узел типа И-ИЛИ. В этой системе присутствует в зачаточной форме "самосознание", поскольку она способна управлять своим поведением, запоминая новые правила и применяя их к самой себе. Бытует анекдот, что система обнаружила факт, что правила, сформулированные человеком, лучше ее собственных, и создала новое правило:

ЕСЛИ правило создала машина ТО вычеркни его. ...И первым делом стерла это правило.

Что касается основной массы систем, использующих автоматическую генерацию новых знаний, они более или менее далеки от EURISKO и относятся к одной из следующих нескольких категорий.

1. Обучение с выводами низкого уровня сложности.
2. Параметрическое обучение.
3. Обучение по аналогии.
4. Обучение на основе индукции — эвристическое обучение.
5. Обучение с использованием специфических методов.

Примером обучения с выводами на низком уровне может служить TEIRESIAS — система-консультант в области медицины, разработанная на базе системы MYCIN. Специалисты-медики являются преподавателями обучающей системы. Система поддержки знаний изучает ответы на поставленные задачи и корректирует те правила в БЗ, которые ранее приводили к ошибкам. Для ввода знаний используется встроенный редактор базы знаний.

Следующие три типа обучающих стратегий можно объединить под общим названием "обучение на примерах". При реализации этой стратегии происходит сбор фактов, их обобщение и использование в качестве новых знаний.

Параметрическое обучение состоит в определении общего вида решающего правила и корректировки входящих в это правило параметров в соответствии с исходными данными. Примером такой системы является META-DENDRAL.

Выводы по аналогии — одно из важных направлений приобретения знаний. Специалисты рабочей группы по междисциплинарному подходу к человеческим факторам в вычислительных системах (COST-11) считают, что процесс появления и изменения структур в семантической памяти человека основывается на аналогиях: с новой ситуацией связываются известные понятия и структуры. Теоретическое описание этого подхода приведено в [3].

Вывод по индукции осуществляет обобщение совокупности имеющихся данных для получения общего правила. Такое обучение включает также создание новых структур, построение теорий, открытие новых правил. Этот тип обучения реали-

зован в системах MARVIN, NODDY, BACON, COPER. Компанией Radian Corporation разработана и предъявлена в качестве коммерческой система RULEMASTER — инструментальное средство, облегчающее разработку основанных на правилах экспертных систем. Ее важнейшая особенность — индуктивный вывод правил из наборов примеров, иерархическая организация порожденных правил.

Наибольшее впечатление производит группа обучающихся программ, названных так за свою способность порождать новые правила высокого уровня значимости, названная системами "обучения открытиям". Практически на настоящий момент можно назвать две реальные системы, претендующие на принадлежность к этой группе. Это вышеуказанная система EURISKO и разработанная тем же автором (Д.Ленатом) система AM — самостоятельная, "открывшая" интересные закономерности в области элементарной математики.

Недавно появился новый интересный класс обучающихся систем, основанных на так называемых генетических алгоритмах [1]. Эти алгоритмы базируются на стандартных моделях наследственности и эволюции из области популяционной генетики.

Отметим, что в работе [3] имеется прекрасное изложение теоретических основ обучения на примерах. Для характеристики сферы практических приложений обучающихся систем приведем слова Р.Форсайта: "...следует сразу же отсоветовать вам брать программу EURISRO в качестве модели для собственной системы, если только ваша комната не забита специализированными LISP — машинами и у вас масса свободного времени". Наличие LISP- машин (одного вида символьных ЭВМ, для которых внутримашинным языком является система команд языка высокого уровня LISP), реализующих архитектуру ЭВМ будущего, говорит о том, что EURISKO пока что выглядит попыткой заглянуть в будущее.

Таким образом, мы рассмотрели два из указанных выше подходов к процессу приобретения знаний. Оба подхода — непосредственный диалог с экспертом и самообучение компьютерных программ — объединяет невозможность на данном этапе преодолеть трудности, связанные с "человеческим фактором", будь то учет психологических особенностей конкретного эксперта или моделирование процесса рассуждений человека. Возможен ли другой путь, например, в диаметрально противоположном направлении? Может быть, не надо преодолевать психологические особенности человека, а сотрудничать с ними и на этой основе строить процесс приобретения знаний?

Построение индивидуальной модели представления проблемной ситуации экспертом

Теория конструктов как раз представляет собой удобный подход для понимания теоретических проблем проникновения во внутренний мир человека.

Дж.Келли

В 1955 г. в США вышла в свет книга Дж.Келли "Психология личных конструктов", где излагались оригинальная теория личности и новый метод исследования личности. Дж.Келли — высокоэрудированный человек, талантливый врач-психотерапевт, получивший прекрасное математическое образование, — создал теорию личности, которую переводчики наградили названием "теория личных (или личностных) конструктов", и разработал методику ее практического применения — технику "репертуарных решеток". Теория и метод, предложенные Дж.Келли для врачебной психотерапевтической практики, получили новый импульс для их практического использования в последние годы в связи с бурным внедрением персональных компьютеров в различные профессиональные области, с появлением экспертных систем, в первую очередь медицинских. В этих условиях получили

неожиданное направление их применения в области экспертных систем, а именно в процессе приобретения знаний от эксперта. Дадим краткую характеристику этому в высшей степени своеобразному, интересному и многообещающему методу. Единственным его описанием на русском языке является книга [6].

В основе теории конструктов лежит идея о том, что каждый человек представляет собой исследователя. Келли полагал, что любой человек в течение всей жизни ищет смысл в себе, в окружающей обстановке, Вселенной. Для этого он создает и перестраивает свои собственные системы взглядов, выдвигает гипотезы, проверяет их на практике, корректирует, вносит изменения в теорию (изменяет свое мышление) и так до бесконечности. Келли разработал технику репертуарных решеток в качестве метода изучения систем личных конструктов другого человека, как способ "влезть в шкуру другого человека", увидеть мир его глазами, а также, по выражению Келли, "увидеть то, что стоит за словами", т.е. выявить те самые глубинные семантические связи внутри проблемной области, которые эксперту практически невозможно вербализовать.

Репертуарная решетка представляет собой матрицу, которая заполняется либо самим испытуемым, либо экспериментатором в процессе беседы. Столбцам матрицы соответствует группа элементов, объектов — значимых понятий из конкретной предметной области. Строки матрицы представляют собой конструкты — биополярные признаки, параметры, шкалы. Конструкты либо задаются исследователем, либо выявляются с помощью специальных процедур.

Понятие конструкта — центральное в теории Келли. Он заметил, что когда человек говорит: "что-то является чем-то", он всегда неявно предполагает, чем это что-то одновременно не является. Именно конкретное расположение оценки данного элемента на оси "является чем-то" — "не является чем-то" определяет индивидуальное восприятие этого элемента конкретным человеком. Когда мы говорим: "Билл честен", мы не имеем в виду, что Билл обладает этим качеством вследствие того, что он не хризантема, не линейный крейсер, не квадратный корень из минус единицы. Говоря, что Билл честен, мы подразумеваем только то, что он не плут. Степень честности Билла мы неявно оцениваем местоположением на оси "честен-бесчестен". Совокупность конструктов представляет собой набор значимых осей, относительно которых человек рассматривает и оценивает свой и окружающий мир. В процессе заполнения репертуарной решетки испытуемый оценивает каждый элемент по каждому конструкту. Заполненная решетка подвергается статистической обработке вручную или на ЭВМ. Таким образом, репертуарная решетка — есть "окно в мир другого человека". Определение "репертуарная" означает, что элементы выбираются по определенным правилам, соответствуют определенной области и связаны между собой осмысленным образом (контекстом исследуемой проблемной ситуации). Анализ репертуарной решетки позволяет оценить силу и направленность связей между конструктами, выявить наиболее значимые конструкты (глубинные), а также иерархические отношения между конструктами. На рис. 2 приведена заполненная оценочная решетка (для оценки используется 11-балльная шкала), позволяющая сравнить между собой восемь различных теорий исследований личности по восьми параметрам, значимых для обоснования теорий. Элементы этой решетки — фамилии ученых, авторов теорий, конструкты — теоретические положения (пример взят из книги [6]).

Обработка этой решетки с помощью статистического пакета программ и анализ интеркорреляций конструктов (рис.3) позволил авторам этой решетки прийти к выводам типа: теории, делающие акцент на свободе

	Фрейд	Эриксон	Мюррей	Скиннер	Олпорт	Келли	Маслоу	Роджерс	
Свобода	11	8	10	11	5	6	2	1	Детерминизм
Рациональность	10	3	2	6	1	1	2	1	Иррациональность
Холизм	3	1	3	11	3	3	1	1	Элементаризм
Наследственность	3	10	6	11	6	9	5	4	Окружающая среда
Субъективность	5	8	4	11	5	1	2	1	Объективность
Активность	4	3	4	11	1	6	1	1	Реактивность
Гомеостаз	1	9	2	6	10	6	10	11	Гетеростаз
Познаваемость	1	4	6	1	4	11	11	11	Непознаваемость

← "1" ——————————————————— "11" →

Рис.2

человека в противоположность детерминизму, не имеют в своей основе допущений о "познаваемости" и "гомеостазе" (интеркорреляция соответствующих элементов отрицательна), однако они базируются на представлениях о "субъективности", "рационализме" и "активности" (высокие интеркорреляции).

Метод репертуарных решеток сейчас — это целое направление в практической психодиагностике. Имеются исследования артикулированности (степени структурированности и связности) систем конструктов, показавшие, например, качественные различия в решетках, заполненных здоровыми людьми и больными неврозами. Интенсивность — другой показатель, позволяющий по данным решетки дифференцировать людей по степени "рыхлости" системы конструктов. "Рыхлая" система конструктов свидетельствует о том, что человек в данной области не способен четко мыслить и планировать свои действия.

Есть и другие показатели, имеющие к обсуждаемой теме (эксперт и его знания) непосредственное отно-

Свобода/детерминизм	X	0,69	0,58	0,34	0,69	0,68	0,82	-0,82
Рациональность/иррациональность	0,69	X	0,40	-0,13	0,49	0,42	-0,64	-0,73
Холизм/элементаризм	0,58	0,40	X	0,54	0,70	0,90	-0,30	-0,56
Наследственность/окружающая среда	0,34	-0,13	0,54	X	0,61	0,67	0,10	-0,21
Субъективность/объективность	0,69	0,49	0,70	0,61	X	0,62	-0,18	-0,85
Активность/реактивность	0,68	0,42	0,90	0,67	0,62	X	-0,46	-0,45
Гомеостаз/гетеростаз	-0,82	-0,64	-0,30	0,10	-0,18	-0,46	X	0,48
Познаваемость/непознаваемость	-0,82	-0,73	-0,56	-0,21	-0,85	-0,45	-0,48	X

Рис.3

шение (например, когнитивная сложность — способность оценивать внешний мир одновременно по определенному количеству параметров и др.).

Таким образом, по сути, теория личных конструктов является тем методом, который ориентирован на решение задачи приобретения знаний у эксперта с учетом его собственного, специфического, субъективного видения проблемной ситуации. Одновременно метод позволяет оценить уровень соответствия испытуемого тем требованиям, которые предъявляются к эксперту. По всей видимости, решетку можно рассматривать как разновидность структурированного интервью. Обычно мы неявно исследуем систему конструктов другого человека в ходе разговора с ним, начиная в процессе беседы понимать, как он видит мир, как оценивает ситуации и т.п. Решетка формализует этот процесс и дает математическое обоснование связей между конструктами.

Эти факты не остались вне поля зрения сотрудников фирмы Boeing, которые с начала 80-х годов занялись разработкой коммерческих систем для приобретения знаний у человека-эксперта с их последующим объединением в БЗ системы и получением "быстрого" первоначального прототипа — целевой ЭС в различных областях. Время, затрачиваемое при этом на создание "быстрого" прототипа, — порядка двух часов.

Средства приобретения знаний на основе теории личностных конструктов

Для любого из нас проникновение во внутренний мир другого означает реконструкцию опыта этого человека, и бессмысленно надеяться, что он сумеет просто передать нам свой опыт, будто открыть карты?

Дж.Келли

Одно из дополнительных преимуществ, которыми обладает техника репертуарных решеток по сравнению с традиционным диалогом, — наличие своеобразной процедуры выявления конструктов методом диад (попарное сравнение элементов и выявление значимых сходств и различий между ними) или триад (сравнение троек элементов). Эта процедура прямо-таки создана для ее автоматизации, поскольку ориентирована на организацию целенаправленного диалога с экспертом, исключающего зацикливания и тупиковые ситуации, сопровождающиеся тягостным молчанием обеих сторон, как случается при традиционном интервьюировании. В дополнение к методу диад и триад разработана "лестница" — методика выявления суперординарных (более высокого уровня абстракции) и субординатных (подчиненных) конструктов с помощью вопросов "как" и "почему", что позволяет выявлять знания с учетом их иерархической структуры. Прелесть непринужденного и заинтересованного разговора с экспертом в рамках технологии выявления конструктов автор имел честь ощутить сам. Особо яркими эти впечатления становятся на фоне воспоминаний о попытках выудить у эксперта крупинки того бесценного глубинного знания, которым он обладает и которое обычно выражается словами: "Я сразу вижу, что это так, но почему это так, сказать трудно".

Итак, родоначальницей целого семейства подобных систем (см.[7]) была разработанная Дж.Боузом ETS-система передачи знаний, построенная на основе KS-300tm, что в свою очередь является расширенной версией MYCIN. ETS обеспечивает инструментальные средства для существенного сокращения процесса приобретения знаний и улучшения качества выявляемых знаний. ETS интервьюирует эксперта, помогает определить набор элементов, сконструировать первоначальный набор эвристик и параметров проблемы. Например, при создании БЗ для консультативной экспертной системы, советующей разработчикам программного обеспечения, какую СУБД использовать для прикладной проблемы, элемен-

тами будут наименования доступных для использования СУБД.

Конструкты вначале выявляются методом триад. При этом ETS предлагает эксперту три СУБД и предлагает подумать о некоторой важной характеристике, которой две из предложенных СУБД обладают, а третья — нет. После того как эксперт укажет имя характеристики, его просят оценить все три СУБД по пятибалльной шкале, где "5" означает наибольшее сходство с указанной характеристикой. После этого эксперту предъявляется следующая тройка элементов и процесс продолжается. Примеры конструктов, выявленных у эксперта в области СУБД: "запускается по ЭВМ типа VAX — не запускается на VAX ", "иерархическая — не иерархическая", "требуется высокий уровень опыта — достаточен низкий уровень опыта", "хорошая диагностика — диагностика отсутствует", "есть поиск — нет поиска".

Если эксперт испытывает затруднения в режиме триад, ETS вводит более простой режим выявления конструктов методом диад. Результатом первой фазы интервьюирования является список классифицированных элементов и параметров классификации, полученных целиком и полностью от эксперта абсолютно без всякого принуждения и навязывания чужой точки зрения, без чего вряд ли сможет обойтись живой инженер по знаниям. Для введения иерархических связей в систему конструктов дополнительно применяется "лестница". Заполненная решетка обрабатывается с помощью методов непараметрического факторного анализа на предмет выявления устойчивых скоплений конструктов. Анализ скоплений выделяет так называемые констеллятивные конструкты, которые функционально близки заданным элементам. Кроме того, как мы помним, характер структурированности конструктов позволяет делать выводы о способности эксперта мыслить четко и последовательно. ETS осуществляет анализ противоречий в системе конструктов, анализируя двойственные отношения между конструктами с помощью графа импликаций. Дуги графа передают маршруты в системе конструктов эксперта. Этот граф предлагается вниманию эксперта. Обычно первым ощущением эксперта при этом бывает удивление множеству связей, которое обнаруживает граф. Второе ощущение — несогласие с наличием или отсутствием некоторых дуг. Граф редактируется, система конструктов перегенерируется. Если противоречия не удается удалить простым редактированием, значит причина противоречия кроется более глубоко. Один из спорных конструктов разбивается на два или более новых и обрабатывается простейшим "лестничным" методом. Возможна ситуация, когда в результате всех ухищрений системы двойственность все же остается, и эксперт в конце концов приходит к выводу, что граф правильно отражает зависимость в предметной области, правда, он никогда раньше не думал об этой характеристике проблемы таким образом.

Двойственные отношения и дуги, с которыми эксперт категорически не согласен, являются важнейшими описателями структуры знаний эксперта, так как представляют собой точки противоречий в методах решения задач экспертом. Процесс разрешения этих противоречий вызывает "психологическое движение, интуицию", по выражению Хинкла, разработавшего вопрос о связи двойственных отношений и конфликтов. Здесь мы встречаемся еще с двумя замечательными свойствами ETS:

- умение ненавязчиво заставить эксперта посмотреть на проблемную область с новой, подчас неожиданной для него точки зрения; это гораздо лучше, чем заставлять его искоренять противоречия;
- влияние на усиление мотивации, заинтересованности в успехе работы в качестве эксперта.

На следующем этапе работы ETS генерирует два типа решающих правил: промежуточные (на основании графа импликаций) и заключающие

(на основе индивидуальных оценок решетки). После чего ETS применяет эвристический алгоритм генерации факторов уверенности для каждого правила.

Распечатки и сообщения, генерируемые системой, полезны в последующей фазе главного интервьюирования, проводимого инженером по знаниям. К этому моменту имеется сформированный базисный словарь, важные характеристики проблемы и область противоречий, по поводу которой сразу же может быть начато обсуждение.

После построения шести прототипов ЭС с помощью ETS были сделаны следующие выводы:

- использование ETS экономит примерно два месяца из пяти, которые уходят на работу инженера по знаниям методом традиционного диалога;
- "искусная" ЭС должна строиться с использованием ETS в комбинации с более традиционными методами и средствами (например, TEIRESIAS — для совершенствования БЗ, сформированной с помощью ETS; META-DENDRAL и AQ11 — для анализа обучающих примеров из БЗ системы, построенных в соответствии со сгенерированными правилами, с использованием индуктивного вывода и т.п.);
- трудно применять классическую методологию решеток для выявления глубинных причинно-следственных связей, процедурных знаний или знаний стратегий, хотя уже имеются некоторые исследования в этих направлениях;
- трудно проверить, что выявлен достаточный набор конструктов.

В заключение отметим, что система ETS реализована в среде InterLisp на ЭВМ Xerox 1100 Dolphin Lisp Machine.

Разработанная несколько позже тем же автором система AQUINAS является расширенной версией ETS и поддерживает следующий набор задач приобретения знаний:

- выявление различий;
- декомпозиция проблем;
- комбинирование неточной информации;
- расширенное тестирование;
- интеграция типов данных;
- автоматическое расширение и модернизация БЗ;
- использование множественных источников информации.

Для реализации последнего пункта разработан механизм выявления знаний из нескольких экспертов для одной предметной области, согласования мнений экспертов, объединения знаний в единую БЗ. Организация режима работы ЭС с пользователем предполагает, что пользователь имеет возможность выбрать, мнения каких именно экспертов из всего состава он желал бы учитывать, получить рекомендации, основанные на мнении большинства, мнении консилиума (мнения, удовлетворяющего всех), мнении.

Таким образом, применение техники репертуарных решеток для приобретения знаний у человека-эксперта обладает бесспорными достоинствами:

- быстрое создание прототипа ЭС и гибкость анализа ее работы;
- эксперт в работе с предметной областью использует свою собственную систему взглядов, лексикон и методы решения, являясь абсолютно свободным от психологического давления со стороны собеседника;
- выявление удовлетворительного количества допустимых решений и их особенностей;
- интерактивная проверка и модификация знаний в процессе их приобретения;
- выявление точек противоречия в методах эксперта;
- повышение мотивации эксперта;
- возможность создания оболочки ЭС, ориентированной на гораздо больший круг проблем, чем традиционные оболочки.

Энтузиазм по поводу скорейшего внедрения рассмотренной техники приобретения знаний, может быть, немного остудит тот факт, что замечательные ситемы, подобные ETS и AQUINAS, реализованы на специали-

зированных символьных ЭВМ. Имеется, однако, коммерчески доступный отечественный пакет KELLY, ориентированный на IBM PC и реализующий классическую схему методики репертуарных решеток. Он дает хорошее представление о сущности, возможностях и ограничениях метода репертуарных решеток. У теории личных конструктов Дж.Келли трудная судьба, сопровождающаяся периодами забвения и поворотами в неожиданные сферы человеческой деятельности. Эта изначально-психотерапевтическая методика формирования индивидуального портрета личности, оживив процесс приобретения знаний у экспертов в конкретной предметной области, должна придать импульс развитию автоматизированных систем обучения (АСО). Для качественного решения задачи обучения (в рамках высшей школы, например) необходимо приобретение и комбинирование в единое целое знаний как минимум трех типов: знания об изучаемой предметной области (аналогично традиционной БЗ); знания о педагогических приемах и стратегиях обучения (область педагогики); о психологических особенностях личности, характеристиках мыслительной, познавательной деятельности (область психологии). Применение теории личностных конструктов здесь имеет широчайшие перспективы.

Теория личностных конструктов и автоматизированные системы обучения (АОС)

...Ведущие направления, определяющие течение психологических процессов, каждый человек вычерчивает для себя сам. Эти путеводные нити — его индивидуальные конструкты... Имея такие конструкты человек может войти в мир реальности и действовать в нем, проявляя инициативу и оригинальность.

Дж.Келли

Процесс создания сценария будущей АОС преподавателем — процесс творческий и потому должен быть освобожден от всяческих рутинных операций (типа разбиения материала на порции, установления порядка следований порций) для максимальной сосредоточенности на элементах и отношениях предметной области. Этот процесс представляет собой не что иное, как приобретение знаний у эксперта-преподавателя, являющегося дважды экспертом: в профессиональной области и в области педагогики. В рамках теории Келли разработаны также методы сравнения нескольких решеток в смысле статистической близости. Налицо все факты использования метода репертуарных решеток. Применение методики репертуарных решеток позволяет выявить единую природу сложно структурированных знаний трех типов, используемых в АОС: предметные; методико-педагогические, психологические, учитывающие индивидуальность обучаемого. Из особенностей используемого метода следует непосредственно обучающий характер диалога АОС со студентом, акцентирующий внимание студента на особенностях элементов и их зависимостях друг от друга, что и является непосредственной целью обучения. Предварительное ранжирование решетки преподавателем позволяет организовать обучение в соответствии со степенью важности очередной порции информации, связанной с текущим элементом. Возможность использования как заданных, так и выявленных элементов и конструктов позволяет управлять мотивацией познавательной деятельности. Так, режим выявления целесообразно использовать для обучения продвинутых студентов, а также тех, чей психологический портрет указывает на наличие тенденции преодоления познавательных барьеров.

Использование метода личностных конструктов позволяет передавать студенту опыт профессиональной деятельности преподавателя. Этот личный опыт проявляется также и в поясняющей информации, которая легко структурируется и детализируется относительно конструктов преподава-

теля; она автоматически выявляется при помощи хорошо разработанных приемов типа лестницы. Индивидуализация обучения для каждого студента также уже заложена в самой сути метода и заключается в том, что каждый студент может исследовать предметную область в соответствии с той стратегией, которую он считает целесообразным применить. Система не навязывает ему своей стратегии и так преодолевается одно из "узких мест" современных адаптивных АОС. Структура решетки определяет легкость реализации гибкой подсистемы объяснений, генерирующей ответы на запросы типа "почему?".

Главной целью реализации АОС является обучение и оценка текущего уровня знаний студента относительно уровня знаний преподавателя. Сравнение двух решеток (эталонной, отражающей представления преподавателя, и решетки, заполненной обучаемым в ходе диалога) позволяет оценить различия в представлениях преподавателя и обучаемого. Разработанная Слейтером программа DELTA позволяет анализировать распределение элементов в двух решетках, подсчитывать корреляции между конструктами двух решеток, между решетками в целом, проводит обработку матрицы различий между двумя решетками и другие виды анализа отношений между решетками.

Использование статистических методов сравнения решеток предъявляет достаточно жесткие требования к организации диалога в АОС в силу необходимости выполнения определенных условий для проведения статистической обработки. Возможно предоставить обучаемому свободу в выборе методов решения (так называемый режим свободного обучения). Тогда задача сравнения двух решеток может рассматриваться как задача распознавания (несходства) двух нечетких образцов. Для ее решения можно использовать индексы сравнения двух нечетких чисел [8], позволяющие определить, какое из двух нечетких чисел больше другого.

Возвращаясь к начальному тезису статьи о том, что инженерия знаний как научное направление находится в процессе становления, на начальном этапе развития, где отсутствуют хорошо зарекомендовавшие себя со всех сторон в процессе длительной эксплуатации методологии, можно констатировать, что этот начальный этап выглядит обещающе. Существуют по крайней мере три рассмотренных подхода к проблеме приобретения и накопления новых знаний, по каждому из которых проведена солидная теоретическая проработка, имеется целый спектр различных практических реализаций: от экзотических исследовательских прототипов до коммерческих систем. Следующим, очевидно, будет решаться такой вопрос: в виде какого сногсшибательного заключения проявится страсть к оригинальничанью у системы EURISKO, если ее интеллектуальный потенциал объединить со способностью тонко исследовать структуру знаний, присущую системе AQUINAS?

Литература

1. Экспертные системы. Принципы работы и примеры/Под ред. Р.Форсайта. — М.: Радио и связь, 1987.

2. Уотермен Д. Руководство по экспертным системам. — М.: Мир, 1989.

3. Приобретение знаний/Под ред. С.Осуги, Ю.Саэки, — М.: Мир, 1990.

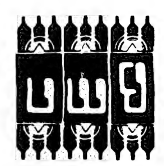
4. Lenat D. EURISKO: a program that learns new heuristics and domain cjncepts 33 Artificial Intelligence. — 1982. — P.21.

5. Проблемы проектирования экспертных систем//Тезисы докладов Всесоюзной школы-совещания — М.:1988. — Ч.2. — С.268-270.

6. Франселла Ф., Баннистер Д. Новый метод исследования личности. — М.: Прогресс, 1987.

7. Tohn H. Boose. Uses of repertory grid-centred knowledge acguisition tools for knowledge-based system//Int. T.Man-Machine Studies. — 1988. — 29. — Pp.287-310.

8. Дюбуа Д., Прад А. Общий подход к определению индексов сравнения в теории нечетких множеств//Нечеткие множества и теория возможностей/Под ред. Р.Ягера. — М.: Радио и связь, 1986. — С.9-21.



Чем сложнее объект той или иной науки, становящейся лидером, тем позднее он может быть изучен в своих элементарных формах и соответственно этому тем дальше от начала общего ряда наук находится его место.

Б.М.Кедров

А.Л.Ездаков

Экспертные системы! Это просто?!

При решении практических задач во многих областях человеческой деятельности бывает затруднительно, а то и просто невозможно использовать формализованные, теоретически обоснованные методы. Так, например, в медицине при установлении диагноза заболевания пациента и назначения курса лечения необходим так называемый индивидуальный подход. Что же представляет собой этот индивидуальный подход? В первую очередь это означает определение зависимостей между показателями общего состояния организма, а именно: наследственность, возрастные изменения, ранее перенесенные заболевания, наличие вредных привычек, среда обитания и многие другие факторы — с одной стороны, и симптоматикой и реакцией на лечебное воздействие — с другой. Можно ли построить формальную модель протекания заболевания? Можно, но для этого нужно иметь полное описание конкретного человека; объединяя его с общими характеристиками болезни, можно получить общую модель данной болезни, протекающей у конкретного пациента. В итоге получается задача гигантской размерности, решить которую не сможет даже современная супер-ЭВМ, а до конца формализовать — ни один коллектив математиков и программистов. Аналогичная ситуация возникает при решении многих задач, связанных со сложными системами, как биологическими, так и техническими.

Тупик? Однако известно, что на практике высококвалифицированные врачи успешно справляются с описанной выше медицинской задачей. В чем же дело? Как они это делают? Эти вопросы являются едва ли не основными для разработчиков систем искусственного интеллекта. Основная проблема при разработке систем ИИ — моделирование умственной деятельности людей при решении ими сложных задач из разных областей человеческой деятельности. В частности, одним из направлений в создании СИИ является разработка систем, имитирующих действия и рассуждения человека-специалиста в узкой предметной области, называемого экспертом. Появление таких систем, называемых экспертными (ЭС), ознаменовало переход из сугубо теоретической области искусственного интеллекта (ИИ) в прикладную область. ЭС стали коммерческим продуктом на рынке прикладного программного обеспечения (ППО) в силу своей полезности при решении целого ряда задач, не решаемых ранее в рамках формальных процедур. Об этих системах и будет идти речь.

В конце 60-х — начале 70-х годов сформировалась концепция описания работы специалиста-эксперта, основанная на апостериорных знаниях — эвристиках. Эвристические правила, которыми пользуется эксперт при решении сложной задачи, формируются в подавляющем большинстве случаев в процессе многолетней интенсивной работы в данной предметной области и являются элементами процесса вывода решения. Поэтому к одной из сложнейших задач построения ЭС относится задача построения базы знаний (БЗ) системы, в решении которой выделяются два этапа: разработка структуры БЗ и заполнение ее.

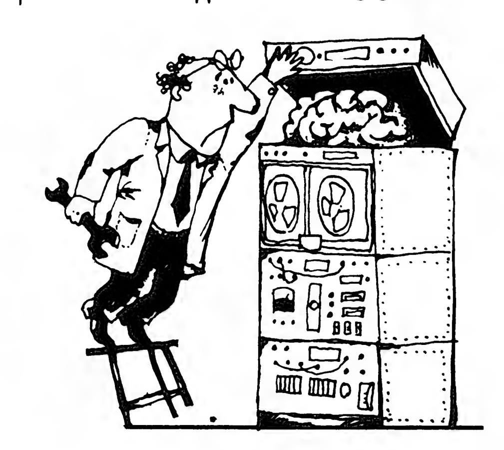
При работе возможно возвращение со второго этапа на первый с последующим изменением структуры БЗ в связи с получением у эксперта знаний, не укладывающихся в структуру БЗ, выбранную первоначально. Таким образом, процесс формирования БЗ приобретает итеративный характер. Одним из наиболее сложных моментов процесса формирования БЗ, вызывающем итеративные возвращения, является получения знаний у эксперта с их последующей формализацией. Поскольку многие эвристики сам эксперт считает очевидными, а некоторые использует интуитивно, бывает очень трудно получить от него связный, логически последовательный рассказ о процессе решения той или иной задачи. В силу этого не каждый специалист в предметной области может быть экспертом. В целом в этой области разработки ЭС возникает ряд сложностей психологического характера, среди которых имеются и такие, как нежелание и неумение специалиста сформулировать свои знания в допускающей формализацию форме.

Вторая важнейшая часть любой ЭС — машина логического вывода (МЛВ), к основной задаче которой относится согласованная обработка данных, имеющихся в программной среде или полученных в процессе диалога с объектами внешней среды. При этом используются знания, хранящиеся в БЗ, с целью получения конечного результата. Под внешней средой подразумевается либо некоторая техническая система, информацию о которой ЭС получает посредством датчиков, либо человек-оператор, работающий с ЭС. Полученные таким образом данные анализируются или интерпретируются с помощью хранящихся в БЗ знаний эксперта. В итоге:

- выдвигаются и проверяются различные гипотезы;
- вырабатываются новые данные, а в некоторых случаях и новые знания;
- формируются запросы на ввод новых данных;
- формируются решения, носящие рекомендательный или управляющий характер, в зависимости от типа ЭС.

Для большинства ЭС важной частью является подсистема объяснения (ПСО). Основная цель этой подсистемы — сделать ЭС "прозрачной" для пользования, т.е. предоставить пользователю возможность понимать логику действий системы. Эта возможность имеет большое значение в процессе освоения ЭС пользователем, когда недоверие к системе, существующее на первом этапе работы, сменяется пониманием и в конечном итоге ведет к полному приятию системы пользователем. Особенно это важно для обучающе-консультирующих ЭС, где от отношения человека к системе в психологическом аспекте полностью зависит успех работы. Подсистема объяснения должна позволять пользователю:

- в любой момент приостанавливать работу системы и получать полное описание ее текущего состояния;
- по запросу пользователя выдавать любую информацию о пройденном системой пути с возможностью возврата на любой его участок;
- по запросу пользователя сообщать результаты ранее выполненных действий, проверок выдвинутых гипотез с пояснениями;
- получать ответы на вопросы типа "почему?", "зачем?", "как?". Надо отметить, что существует ряд ЭС (прежде всего управляющих), для которых наличие развитой ПСО не обязательно. Тем не менее в целом роль такой подсистемы в ЭС важна.



Еще одной необходимой составной частью ЭС является редактор базы знаний (РБЗ). Основная задача РБЗ является предоставление инженеру по знаниям средств, облегчающих процесс заполнения БЗ в процессе работы с экспертом. Очевидно, что БЗ имеет вид набора файлов, хранящихся на магнитном носителе. Следовательно, такой редактор представляет собой транслятор с некоторого подмножества естественного языка, используемого инженером по знаниям, в специальный код, ориентированный на работу МЛВ. Редактор базы знаний должен:

- быть удобен для работы инженера по знаниям, т.е. использовать некоторое подмножество естественного языка;
- позволять реализовывать любую структуру знаний, предусмотренную составом БЗ;
- позволять корректировать элементы БЗ без изменения ее структуры;
- иметь возможность представлять обобщенную структуру БЗ. К желательным, но не необходимым характеристикам редактора относится возможность проверки на непротиворечивость БЗ и отсутствие в ней зацикливаний в процессе логического вывода. Основным же элементом РБЗ является "дружественный" интерфейс с инженером по знаниям.

Таким образом, мы кратко рассмотрели четыре основные компоненты ЭС. Хотя возможна реализация ЭС без третьей (подсистемы объяснений) и четвертой (РБЗ) компонент, подавляющее большинство существующих и разрабатываемых ЭС обладает всеми в различной степени развитыми компонентами. Так, одна из первых ЭС-MYCIN [1], осуществлявшая диагностику бактериальных инфекционных заболеваний крови, имеет продукционную БЗ, развитую МЛВ, учитывающую вероятность получения правильного результата, развитые ПСО и РБЗ, ориентированные на работу с продукционными правилами. Рассмотрим перечисленные компоненты ЭС и методы их взаимодействия при функционировании реальных систем.

База знаний

В БЗ в некотором закодированном виде хранятся формализованные знания эксперта. На современном этапе развития ЭС используется несколько форм представления знаний. Выделим из них четыре основные:

1) "Тройка" объект-атрибут-значение, например: дом-цвет-зеленый; пациент-температура-высокая. Эта форма представления знаний определяет "объект", обладающий некоторыми атрибутами (свойствами), которые могут принимать значения из известного набора.

2) Правила продукций в виде: Если пациент болен гриппом И стадия заболевания начальная, ТО температура высокая с вероятностью = 0,95 И головная боль есть с вероятностью = 0,8.

Правило продукции состоит из двух частей: посылки (ЕСЛИ) и заключения (ТО), каждая из которых состоит из конъюнкции утверждений более низкого уровня детализации.

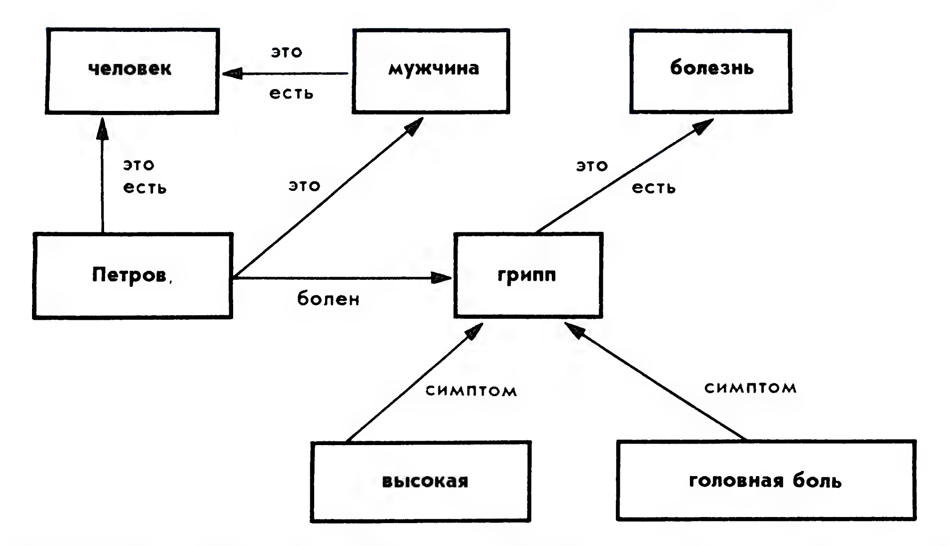
3) Фрейм.

имя:	плащ
слоты	значения
входит в класс	"верхняя одежда"
принадлежит	Петров В.В.
состояние	новый
воротник	стойка
кол-во рукавов	(по умолчанию — 2)
размер	f(R,V,O,X)

Фрейм представляет собой именованную таблицу с некоторым количеством слотов — ячеек, имевших свои имена и получающих в процессе работы машины вывода некоторые значения. В качестве значений могут присутствовать константы, ссылки на фреймы более высокого или более низкого уровня, а также некоторые вычислительные процедуры, например функция f, вычисляющая размер X по росту R, весу V и объему груди O.

4) Семантическая сеть.

Семантическая сеть — это ориентированный граф, вершины которо-



го соответствуют объектам (событиям), а дуги описывают отношения между вершинами.

Первая из указанных форм представления знаний ("тройка") является наиболее ранней формой, своего рода переходной, от представления данных к представлению знаний. Наилучшая область применения "тройки" — диагностические ЭС в предметной области с большим количеством легкокластеризуемых объектов, в каждом классе которых имеется большое количество общих атрибутов. Примером такой области может служить диагностика сложных технических систем.

Представление знаний с помощью правил продукции — самая распространенная форма реализации БЗ [9]. С помощью продукций можно описать практически любую систему знаний. На них основаны все ранние ЭС, такие, как MYCIN, XCON [1] и другие, а также подавляющее большинство современных ЭС в различных предметных областях. В соответствии с этой формой реализации БЗ сделано большинство систем — "оболочек" и многие инструментальные среды. В целом продукционная форма представления знаний является естественной и удобной для формализации знаний, полученных у эксперта.

Фрейм-структуры в определенной степени стали развитием метода электронных таблиц в области обработки знаний. Это вторая по частоте встречаемости в конкретных реализациях ЭС форма представления знаний. Как показано в [6], фрейм-структуры наиболее разумно применять в предметной области с четко выраженной иерархической структурой, например в САПР. В таких областях широко используются некоторые важные свойства фреймов, такие как:

- возможность наследования (или индукции) свойств;
- использование слота типа KEY (ключа);
- широкое применение присоединенных процедур;
- ссылки на фреймы более высокого или более низкого уровня.

Семантические сети изначально возникли как модель долговременной человеческой памяти в психологии. Большой интерес к работе с семантическими сетями проявляют разработчики систем взаимодействия на естественных языках. Однако семантические сети как форма представления знаний успешно использовались в других областях: в медицине (система CASNET [1]), в геологии (PROSPECTOR [1]). Основное преимущество семантических сетей — наглядность — становится недостатком при значительном увеличении объема БЗ [1, 6, 10]. Для преодоления таких трудностей используют метод

иерархизации сетей (выделение на них локальных подсетей) [6], что, однако, не всегда возможно реализовать практически.

На современном этапе работ в области ЭС нередко используется смешанная форма представления знаний в БЗ [2, 8]. Такие ЭС называются гибридными. В гибридных ЭС используются продукционно-фреймовые и фрейм-продукционные модели знаний, а также сети фреймов.

Машина логического вывода

Характерное отличие ЭС от традиционного прикладного программного обеспечения — использование для обработки нового вида информации — знаний. В силу этого МЛВ существенно отличается от алгоритмов, управляющих решением задач в прикладных пакетах. В ЭС выбор формы представления знаний относится к фундаментальному понятию, поэтому механизм вывода определяется в соответсвии со структурой реализации БЗ. Однако существуют и некоторые общие для любых форм представления знаний характеристики МЛВ. К одной из таких характеристик относится направление вывода, которое может быть двух типов: — прямой вывод (от посылок к цели) и обратный вывод (от цели к данным).

Безусловно, прямой вывод более общий, так как он позволяет просмотреть все дерево возможных решений и найти все существующие терминальные вершины. При этом, очевидно, теряется одно из основных преимуществ ЭС — возможность быстрого нахождения решения в большом пространстве поиска решений. В силу этого прямой вывод в чистом виде применяется только в "игрушечных" предметных областях. При работе в реальной области необходимо предусмотреть аппарат отсечения ряда ветвей дерева решения с целью сокращения пространства поиска решения, которое в отдельных случаях может оказаться бесконечным.

Обратный вывод по сути своей служит механизмом проверки гипотез, и поэтому (при условии отсутствия ошибок в БЗ) всегда приводит к решению об истинности или ложности гипотезы за конечное число шагов. Однако здесь возникает вопрос о выдвижении начальных гипотез. Если пространство возможных гипотез велико, как, например, при диагностике заболеваний человека, и отсутствуют начальные установки о приемлемости тех или иных гипотез, то обратный вывод ничуть не эффективнее прямого. На практике при решении задач в реальных предметных областях целесообразно сочетать на разных этапах работы МЛВ прямой и обратный вывод.

Кроме того, машины логического вывода различаются по управлению поиском сначала вглубь, или сначала вширь.

В первом случае вывод решения развивается по некоторой ветви дерева решения до терминальных вершин. Если обнаруженные вершины не будут решением, то берется следующая ветвь (как правило, слева направо), и процесс продолжается в режиме "отката с забыванием" (бэктрэкинга). В данном случае возможно появление "бесконечной ветви", что приводит к недостижимости решения. Но и при отсутствии бесконечной ветви, без учета некоторых ограничений на глубину и последовательности перебора ветвей, получаем ситуацию полного перебора в пространстве поиска решений.

Второй режим управления выводом представляется более разумным, так как обеспечивает обнаружение решения за конечное число шагов. Суть его сводится к последовательному просмотру всех ветвей дерева решения на один шаг в глубину до получения решения. Наиболее целесообразным представляется сочетание поиска вширь с поиском в глубину, заключающееся в просмотре каждой ветви дерева решения не на один шаг, а на некоторое устанавливаемое самой ЭС число шагов. В сочетании с отсечением ряда ветвей по результатам просмотра этот ме-

тод, как и сочетание прямого и обратного вывода, наиболее целесообразен. Необходимо решить вопрос: кто и как будет переключать типы и направления вывода, регулировать глубину просмотра и отсечение ветвей? Для решения этих, а также ряда других вопросов в развитых ЭС существуют мета-правила, которые в соответствии с текущим состоянием системы определяют дальнейшую стратегию работы МЛВ.

Одно из важнейших отличий ЭС от прикладных программ состоит в возможности работы в условиях неполной и неточной информации, являющейся результатом неполных и субъективных данных. Именно поэтому большинство ЭС работает в недетерминированном режиме, т.е. в режиме наличия неопределенности. Для реализации этого режима МЛВ используют различные методы, например:

- методы теории вероятности (в основном, байесовские формулы);
- методы математической статистики (оценка функции правдоподобия и т.п.);
- теорию нечетких множеств Заде [10].

Используются также комбинации этих методов и различные их модификации, оперирующие с понятиями типа "фактор уверенности", "оценки неправдоподобия". В области работы МЛВ нет четкой концепции о применимости и предпочтительности того или иного метода и, можно сказать, царит полный произвол разработчиков ЭС.

По своему характеру логический вывод делится на монотонный и немонотонный.

При монотонном выводе полученные в процессе вывода факты не пересматриваются, т.е. считаются истинными до окончания работы системы. При немонотонном выводе факты не будут чем-то раз и навегда определенным. По мере поступления новой информации факты могут пересматриваться. Немонотонность вывода особенно важно при работе ЭС с динамической внешней средой, особенно при функционировании в реальном масштабе времени [8].

Таким образом, определив структуру БЗ, разработчик имеет возможность в каждом конкретном случае заняться созданием МЛВ, учитывая общие концепции вывода решения, преимущества выбранной структуры БЗ и личный опыт проектирования ЭС.

Подсистема объяснения

Как было указано выше, наличие развитой ПСО чрезвычайно важно для ЭС, работающих в области обучения. В процессе обучения такая ЭС будет выполнять не только активную роль "учителя", но и роль справочника, помогающего "ученику" изучать внутренние процессы, происходящие в системе, с помощью моделирования прикладной области. Развитая ПСО состоит из двух компонент:

- активной, включающей в себя набор информационных сообщений, выдаваемых пользователю в процессе работы, зависящих от конкретного пути решения задачи, полностью определяемых системой;
- пассивной (основной компоненты ПСО), ориентированной на инициализирующие действия пользователя.

Активная компонента ПСО является развернутым комментарием, сопровождающем действия и результаты, полученные системой. Такого рода информационное сопровождение предусматривалось и ранее. Пассивная компонента ПСО — это качественно новый вид информационной поддержки, присущей только системам, основанным на знаниях [5]. Эта компонента, помимо развитой системы HELP-ов, вызываемых пользователем, имеет системы пояснений хода решения задачи. Система пояснений в существующих ЭС реализуется различными способами. Она может представлять собой:

- набор информационных справок о состоянии системы на момент останова;
- полное или частичное описание пройденного системой пути по дере-

ву решений;

- список проверяемых гипотез (основания для их формирования и результаты их проверки);

- список целей, управляющих работой системы, и путей их достижения.

Важной особенностью развитой ПСО является использование в ней естественного языка общения с пользователем. Широкое применение систем-"меню" позволяет не только дифференцировать информацию, но и в развитых ЭС судить об уровне подготовленности пользователя, формируя его психологический портрет. Таким образом, работа ПСО может оказывать влияние на работу МЛВ.

В то же время, как уже указывалось, для ЭС управляющего типа ПСО может быть ограничена только активной компонентой. При управлении сложными техническими объектами в реальном масштабе времени бывает разумным заменить ПСО на систему записи информации о процессе работы МЛВ с последующей расшифровкой в случае необходимости. Это представляется целесообразным с точки зрения управления быстротекущими процессами, в которых значимые для процесса временные интервалы существенно меньше времени реакции человека-оператора.

Редактор базы знаний

Как было указано выше, РБЗ представляет собой транслятор некоторого подмножества естественных языков в файловую структуру записи, позволяющий облегчить работу с ЭС на этапе заполнения БЗ. В принципе РБЗ не обязателен для ЭС. Например, при поставке коммерческого программного продукта РБЗ чаще всего отсутствует. Роль и место РБЗ определяются функциями и структурой заполняемой БЗ. Редактор функционирует в процессе работы с экспертом, в процессе заполнения БЗ инженером по знаниям. РБЗ необходим также в процессе отладки ЭС. Конкретная реализация РБЗ зависит от:

- структуры и объема БЗ;

- средств, выбранных для реализации ЭС;

- задач, ставящихся перед РБЗ;

- личных особенностей разработчиков ЭС.

В зависимости от структуры БЗ РБЗ транслирует знания в различные формы файловых структур. Сами знания должны быть сформулированы существенно различным образом при заполнении, например, БЗ продукционного и фреймового типа. Как будет показано ниже, некоторые средства проектирования ЭС уже имеют встроенный РБЗ или отдельные его компоненты. В зависимости от конкретных требований, разработчики могут наделить РБЗ, помимо функций транслятора, некоторыми аналитическими функциями. Наиболее распространенная из них функция проверки нового элемента БЗ на непротиворечивость с уже имеющимися знаниями. Большой интерес для исследователей и разработчиков представляет реализация функции проверки на отсутствие циклов в работе МЛВ. К отдельной задаче относится проверка БЗ на полноту, что особенно важно для формирования функции логического отрицания в процесе работы МЛВ. Личные качества разработчиков ЭС влияют на все указанные характеристики, но и сами по себе становятся самостоятельным фактором, определяющим, в частости, гибкость интерфейса РБЗ с инженером по знаниям.

Средства разработки ЭС

Средства разработки ЭС можно разделить на четыре категории:

- алгоритмические языки програм мирования (например, широко известные языки программирования такие, как BASIC, FORTRAN, C, PASCAL);

- языки программирования для систем ИИ (наиболее известны среди которых PROLOG, LISP, SMALLTALK);

- инструментальные средства со-

здания ЭС (OPS-5) KEE, ART и другие);

- "оболочки" ЭС (SHELL-системы), которых в настоящее время достаточно много (например, GURU, EXSYS, EMYCIN).

Работая на самом низком уровне, уровне языков программирования алгоритмического типа, можно создать практически любую ЭС. В то же время сроки выполнения такой работы будут велики. Работая же с "оболочками", т.е. с системами, имеющими пустую БЗ, разработчик ограничен строго определенной структурой БЗ, МЛВ, полностью закрытой структурой ПСО и готовым РБЗ. Сроки разработки системы с использованием "оболочки" невелики, однако эффективность работы созданной системы зачастую мала, поскольку, несмотря на обилие различных "оболочек", очень сложно втиснуть исследуемую предметную область в "прокрустово ложе" ограничений, накладываемых любой из "оболочек".

Большую свободу допускают инструментальные средства проектирования ЭС. Они, как правило, позволяют изменять в определенных пределах структуру БЗ системы, настраивать ее на выбранную струкуру РБЗ, комбинировать различные подходы к реализации МЛВ и разрабатывать отдельные части ПСО, используя БЗ. Сроки создания ЭС с помощью инструментальных средств больше, чем с использованием "оболочек", поскольку требуется привлечение к работе, помимо инженера по знаниям, еще и эксперта, и программистов.

Языки программирования систем ИИ отличаются от алгоритмических языков наличием ярко выраженной ориентацией на логическую обработку знаний. В то же время они существенно экономят время разработки при условии правильного их выбора, наиболее полно соответствующего особенностям данной предметной области. К недостаткам языков программирования систем ИИ следует отнести их низкую эффективность на ЭВМ стандартной архитектуры.

В итоге в качестве одной из рекомендаций опытного разработчика ЭС [1, 2, 8] приводится совет начинать разработку с использования "оболочки" или инструментального средства для создания "быстрого прототипа" ЭС для исследования и последующего уточнения концепции. После доведения "быстрого прототипа" до работоспособного и удовлетворяющего поставленным требованиям осуществляется перевод его на язык программирования с целью повышения эффективности работы ЭС. Бесспорный приоритет языков программирования при наличии специальных ЭВМ определяется повышением эффективности этих языков на ЭВМ соответствующей им архитектуры.

Приведем ряд классификаций ЭС, разработанных в странах, активно развивающих эту ветвь систем ИИ. Во-первых, по цели функционирования системы делятся на управляющие, консультирующе-управляющие, консультирующие.

Как было показано, одно из основных различий данных форм работы ЭС заключается в степени развитости ПСО. Во-вторых, по характеру решаемых задач ЭС делятся на анализирующие и синтезирующие. Это различие в определенной степени условно, поскольку большинство систем, работающих в предметных областях, связанных с синтезом решения, содержат в себе этап анализа, по результатам которого формируется синтезируемое решение.

В-третьих, по модели предметные области бывают статические и динамические. В статических предметных областях информация не меняется за один цикл работы системы. В динамических средах происходит обновление модели предметной области в процессе работы ЭС. Для таких предметных областей необходима МЛВ, реализующая немонотонный вывод.

В-четвертых, по сложности структуры ЭС делят на поверхностные и глубинные. Основное отличие глубинных ЭС от поверхностных заключено в способности принимать решение в неопределенных ситуациях, т.е.

способности к самоанализу и самообучению.

И наконец, ЭС делятся по типам используемых знаний на традиционные и интегральные. Традиционные ЭС используют формализованный набор эвристик, полученных у эксперта. Интегральные, помимо набора эвристик, используют обращения к некоторым формальным (процедурным) моделям, позволяя применять наработанное ранее программное обеспечение.

Суммируя сказанное, отметим, что существует разделение ЭС на два больших класса: простые и сложные. Как правило, под простой ЭС понимают поверхнострую, традиционную и предназначенную для использования на ПЭВМ. Объем БЗ такой ЭС обычно не превышает тысячи правил, целью работы является анализ (диагностика) некоторой ситуации. Сложной ЭС называют глубинную интегрированную, ориентированную на символьную ЭВМ рабочую станцию или мощную стационарную ЭВМ.

Таким образом, коллектив, рассматривающий вопрос о разработке ЭС для решения задач в некоторой предметной области, должен ответить на ряд вопросов:

1. Подходит ли такая область для применения ЭС? Здесь необходимо уточнить, не существует ли разработанных алгоритмических подходов, позволяющих решать задачи в данной области? Не является ли предметная область слишком широкой или слишком узкой, позволяющей находить решение методом прямого перебора? И наконец, имеются ли специалисты-эксперты успешно решающие данные задачи в этой области, способные логично и последовательно сформулировать свои мысли о процессе решения задач?

2. Какого рода ЭС необходимо разработать для имеющейся задачи? При решении этого вопроса можно использовать приведенную выше классификацию.

3. Какое количество и каков профиль работы специалистов, которые должны участвовать в работе по разработке ЭС?

4. Какая вычислительная техника имеется в наличии и какая требуется для выполнения указанных работ?

5. Какие средства разработки ЭС доступны группе разработчиков?

6. Каковы объем финансирования и сроки разработки ЭС, предлагаемые руководством или заказчиками?

После получения ответов на эти вопросы необходимо проанализировать ситуацию с точки зрения возможности разработки ЭС. Так, если необходимо разработать ЭС, относящуюся к классу сложных, и при этом выполняется хотя бы одно из условий:

- из предполагаемой группы никто и никогда не работал в области создания ЭС;
- из вычислительных средств имеется в наличии только ПЭВМ;
- среди средств разработки имеется только алгоритмические языки программирования;
- сроки разработки несколько месяцев (до года);
- объем финансирования порядка несколько тысяч рублей.

Разработка ЭС должна быть отложена.

В случае разработки простой ЭС необходимо начать работу с создания "бумажной" ее модели. Эта модель формируется в процессе общения с экспертом. При этом выделяются основные понятия, которыми оперирует эксперт, формируется тезаурус системы. После этого на нескольких несложных примерах подробно анализируется метод, которым эксперт решает такого рода задачи, и делается первая попытка формализации знаний, получаемых у эксперта. На этом этапе рекомендуется использовать продукционную структуру базы знаний. На сформированном первичном наборе продукций проверяется действие МЛВ. Все эти действия осуществляются в рамках "бумажной" модели вручную. На всем этапе в постоянном контакте с экспертом нужно попытаться обобщить продукцию, сделать правила более глобальными и расширить число приме-

ров, обсуждаемых с экспертом.

После проверки и одобрения экспертом "бумажной" модели начинается второй этап, этап переноса модели на ЭВМ. На этом этапе рекомендуется подобрать "оболочку", которая позволила бы полностью реализовать "бумажную" модель в отношении структуры БЗ и МЛВ. После этого происходит наполнение "оболочки" знаниями из "бумажной" модели и предъявление системы эксперту. Таким образом, на втором этапе контакт с экспертом осуществляется только при получении результатов. В случае одобрения экспертом работы системы на ЭВМ происходит переход к третьему этапу. При возникновении у эксперта возражений следует вернуться к "бумажной" модели.

На третьем этапе работы создается "демонстрационный" прототип. Инженер по знаниям в контакте с экспертом заполняет базу знаний системы значительным количеством правил. Обычно считается, что для "демонстрационного прототипа" необходим объем БЗ порядка сотни правил. При заполнении такой базы производится постоянный контроль поступающих знаний на непротиворечивость с уже имеющимся знанием с помощью имеющихся в РБЗ средств или вручную. Следовательно, основную нагрузку на третьем этапе несет инженер по знаниям. Также на этом этапе проверяется успешность выбора эксперта. О степени квалификации эксперта можно судить по проценту банальных заключений системы в общем объеме полученных решений. В процессе работы могут возникнуть сложности с формализацией полученных знаний в соответствии с выбранной структурой БЗ. Все эти моменты фиксируются, но работа должна продолжаться с использованием выбранных средств разработки ЭС. Только в случае возникновения конфликта между возможностями выбранных средств с фундаментальными свойствами знаний, который невозможно ликвидировать без существенных нарушений точности работы ЭС, необходимо вернуться к выбору нового инструментального средства.

В процессе заполнения базы знаний на третьем этапе необходимо постоянно проводить проверку работоспособности системы на тестовых примерах. Основной критерий этих проверок — не ухудшение качества решения тестовых задач с введением нового фрагмента знаний. К важной составной части тестовых примеров относится постоянный контроль "дружественности" интерфейса системы. Поскольку параллельно с процессом заполнения базы, а зачастую и в зависимости от этого происходит формирование ПСО. Постоянный контроль интерфейса позволит развивать эту подсистему в нужном направлении. Фиксирование возникающих при реализации ПСО сложностей позволит устранить ограничения, диктуемые возможностями инструментального средства. Основной целью создания "демонстрационного прототипа" служит показ возможности создания ЭС для работы в данной предметной области на ряде конкретных примеров. Важной задачей, также решаемой в рамках "демонстрационного прототипа", является отладка интерфейса системы с пользователем. Возникшие в ходе работы сложности с реализацией той или иной части ЭС, зафиксированные в протоколах, служат полезной информацией для четвертого этапа работы над системой.

Четвертый этап работы над системой, вообще говоря, не обязателен. Если при работе на третьем этапе не возникло существенных трудностей в формировании всех частей ЭС, то происходит переход к пятому этапу. Если при реализации "демонстрационного прототипа" выяснилась невысокая эффективность системы, связанная с ограничениями инструментальных средств, то происходит возврат ко второму этапу и создание нового "демонстрационного" прототипа с помощью инструментальных средств более низкого уровня, как правило, на универсальном языке программирования. На этом этапе основная нагрузка ложится на инже-

неров-программистов, осуществляющих трансляцию готовой системы с языка одного инструментального средства на другой. После отладки нового "демонстрационного прототипа" происходит переход к пятому этапу.

На пятом этапе происходит разработка "действующего прототипа". Разработка проводится аналогично третьему этапу наполнения базы знаний и ПСО. К важной особенности пятого этапа относится пробная эксплуатация системы на протяжении около полугода. Результаты апробирования системы постояно анализируются с участием эксперта. В результате содержание БЗ должно позволять решать любую задачу из исследуемой предметной области.

На заключительном шестом этапе создается "коммерческая" система. "Действующий прототип" доводится до некоторого уровня, который и эксперт и пользователи считают достаточным, и на него оформляется вся необходимая документация. "Коммерческая" версия системы защищается от несанкционированного копирования и доступа. Устанавливается цена программного продукта, проводятся мероприятия по рекламе.

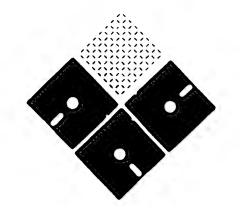
Исходя из такого метода разработки ЭС, коллектив разработчиков должен состоять из четырех-шести человек, в состав которых входят:

- руководитель проекта, осуществляющий научное и административное руководство проектом;
- один-два инженера-программиста, владеющих языками программирования и ориентирующихся в области инструментальных средств и "оболочек" ЭС;
- один-два инженера по знаниям, также имеющих представление о работе с инструментальными средствами и некоторые знания в области психологии общения;
- эксперт-специалист в конкретной предметой области, способный четко сформулировать свои знания и описать процесс принятия решения.

В заключение следует отметить, что, помимо ЭС, существует ряд других направлений в области разработок систем ИИ. В последние годы нарастает количество теоретических и практических работ в области создания нейроподобных структур [11]. Разрабаываемые в рамках данного направления нейрокомпьютеры должны состоять из отдельных элементов, каждый из которых имитирует работу одного нейрона. Такие системы обладают способностью к самообучению на примерах и действуют по принципу сообщества нервных клеток, составляющих человеческий мозг. Энтузиасты нейрокомпьютеров считают, что в отличие от ЭС нейроподобные сети способны решать задачи в любых предметных областях вне зависимости от сложности этой предметной области и от наличия специалистов-экспертов. Данное направление активно развивается и начинает создавать конкуренцию ЭС.

Литература

1. Уотерман Д. Руководство по экспертным системам. — М.: Мир, 1989.
2. Экспертные системы: принципы работы и примеры/Под ред. Р.Форсайта. — М.: Радио и связь, 1987.
3. Хоггер К. Введение в логическое программирование. — М.: Мир, 1988.
4. Вагин В.Н. Дедукция и обобщение в системах принятия решений. — М.: Наука, 1988.
5. Осуга С. Обработка знаний. — М.: Мир, 1989.
6. Представление и использование знаний/Под ред. Х.Уэно., М.Исудзука. — М.: Мир, 1989.
7. Приобретение знаний/Под ред. С.Осуга, Ю.Саэка. — М.: Мир, 1990.
8. Экспертные системы: состояние и перспективы/Под ред. Д.А.Поспелова — М.: Наука, 1989.
9. Кузнецов В.Е. Представление в ЭВМ неформальных процедур. — М.: Наука, 1989.
10. Обработка нечеткой информации в системах принятия решений. — М.: Радио и связь, 1989.
11. Соколов Е.Н., Вайткявичус Г.Г. Нейроинтеллект. От нейрона к нейрокомпьютеру. — М.: Наука, 1989.



Микроэксперт (Micro Expert) был создан братьями Беверли и Уильямом Томпсонами, которые в дальнейшем основали компанию Micro Expert Systems, специализирующуюся в области инструментальных средств искусственного интеллекта для персональных компьютеров.

Т.Г.Дедкова, В.И.Ключко,
В.А.Морозов, А.П.Частиков

РАБОТА С МИКРОЭКСПЕРТОМ

Микроэксперт представляет собой экспертную систему продукционного типа с открытой архитектурой. В ней использован подход, принятый в классической экспертной системе Prospector. Микроэксперт — это так называемая "пустая" экспертная система, которую можно приспособить для решения задач из различных областей знаний, вписывающихся в представление Проспектора. Программа Микроэксперта обеспечивает простой способ изучения, использования и самостоятельного создания экспертных систем. Она иллюстрирует как из правил создается база знаний, как с помощью правил можно получить решение задачи.

Несмотря на то что большинство экспертных систем реализуется на таких языках, как ЛИСП, ПРОЛОГ и им подобным, программа Микроэксперт написана на ПАСКАЛЕ. Язык ПАСКАЛЬ выбран авторами Микроэксперта как наиболее простой язык программирования, обладающий хорошей переносимостью для персональных компьютеров по сравнению с другии языками (в частности с БЕЙСИКом), а благодаря таким недорогим программным продуктам как Турбо-ПАСКАЛЬ, он доступен широкому кругу пользователей IBM PC и совместимых с ними компьютеров "Искра-1030", ЕС 1040 и др.).

Микроэксперт представляет собой набор из девяти файлов: BOTANY.KB, CONIFERS.KB и др. MICRO2.PAS и MICROEXP.PAS — исходные листинги программ Микроэксперта, в которые можно добавлять собственные процедуры и функции пользователя, написанные на языке ПАСКАЛЬ. Программа CRPSSREF.COM предназначена для отладки баз знаний.

Работа пользователя с Микроэкспертом рассматривается на примере базы знаний SAMPLE, что позволит читателю сразу проверить теоретические знания на практике.

Представление знаний в виде правил

Важным этапом создания базы знаний является этап приобретения знаний. Затем разработчику базы знаний необходимо ввести в экспертную систему разнообразный набор фактов о некотором предмете, предварительно представив знания в виде некоторой общей структуры. Одна из таких структур называется "дерево решения". Это не единственный способ представления фактов, и он не может быть применен для всех без исключения задач. Однако дерево решения очень эффективно там, где знания представляются в виде правил.

Структура дерева решения иллюстрирует отношения, которые должны быть установлены между правилами в хорошо организованной базе знаний. Поняв эти отношения, вам будет легче изучить другие организационные методы, может быть, более подходящие для вашей конкретной задачи.

Формализация задачи. Представьте, что вы являетесь специалистом в области ботаники и к вам обратились по телефону с просьбой помочь в определении семейства, к которому принадлежит найденное растение. Так как вы не можете увидеть экземпляр, то вынуждены доверять информации, поступающей от спрашивающего. Однако он не является экс-

пертом и поэтому не знает, какая именно информация вам нужна. Вы вынуждены задавать вопросы, для того чтобы получить сведения, необходимые для решения задачи.

Возможно первое, что вы захотите выяснить, является ли это растение деревом, кустарником, лианой или травой. Начало консультации может выглядеть следующим образом:

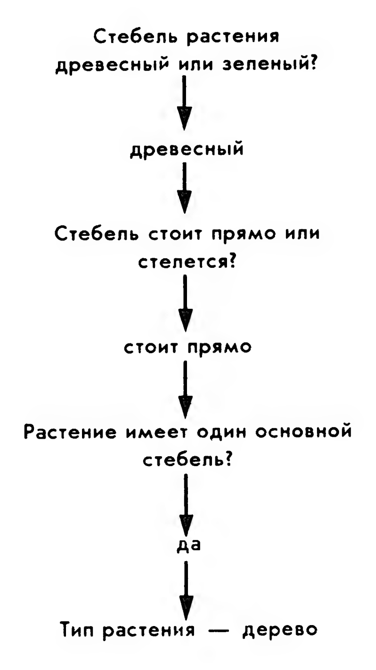


Рис.1. Диаграмма одного из возможных исходов консультации

"Основываясь на ваших ответах, можно предположить, что тип растения — дерево".

Построение дерева решения. На рис.1 показано, как эта телефонная консультация может быть представлена в графической форме.

Данная диаграмма иллюстрирует только один из возможных ответов. Аналогично можно представить и все другие варианты ответов. Такое представление делает консультацию более наглядной. На рис.2 показана диаграмма всех возможных исходов данной консультации. Эта диаграмма и называется деревом решения, которое объединяет все ветви поиска типа неизвестного растения. Вернемся к началу консультации и допустим, что необходимо найти растение, которое является деревом некоторого вида. Для этого надо определить семейство, к которому принадлежит дерево. Следующим этапом будет нахождение класса растения. Класс

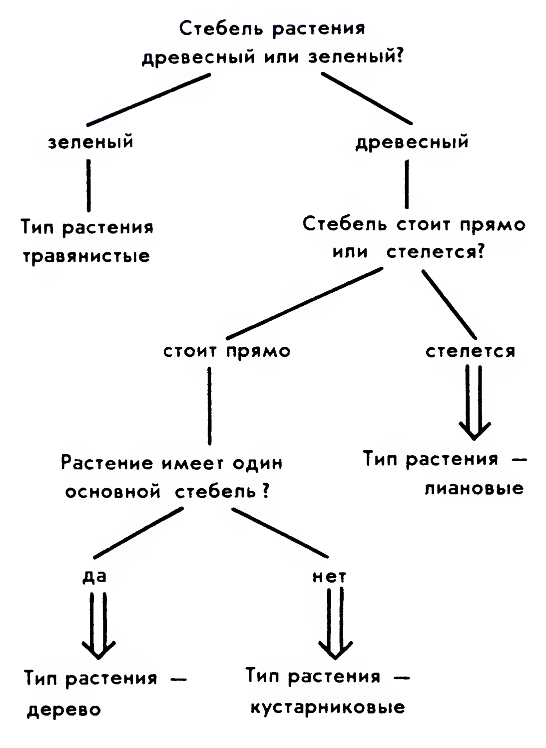


Рис.2. Исходная диаграмма на рис.1 дополнена всеми возможными вариантами ответов

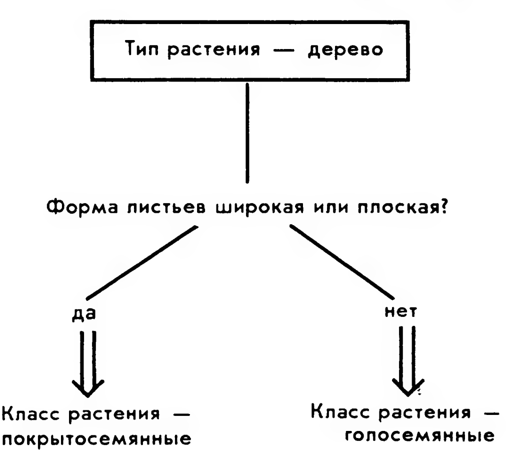


Рис.3. Продолжение дерева для определенного класса растений

можно определить, задав всего лишь один вопрос:

"Форма листьев плоская или широкая?"

"Нет"

Исходя из ответа можно сделать заключение — это класс дерева голосемянные.

Этот вопрос проиллюстрирован на рис.3.

Как видно, новая часть будет "поддеревом" исходного дерева решения на рис.3. Существует несколько причин, по которым все дерево разбивается на секции. Во-первых, это практическая причина: дерево быстро становится длинным и может не поместиться на бумаге, во-вторых, деление дерева решения на секции упрощает запоминание цели, которую мы преследуем, задавая ряд вопросов.

Когда "поддерево" нарисовано, заключительная часть которой копируется в корень вновь созданной ветви дерева решения. В дальнейшем корень будем заключать в рамку. Завершим пример рассмотрением ветви дерева для класса голосемянных. Это дерево решения представлено на рис 4. Структуры, приведенные на рисунке, это незаконченная ботаническая база знаний, которая решает только часть общей задачи.

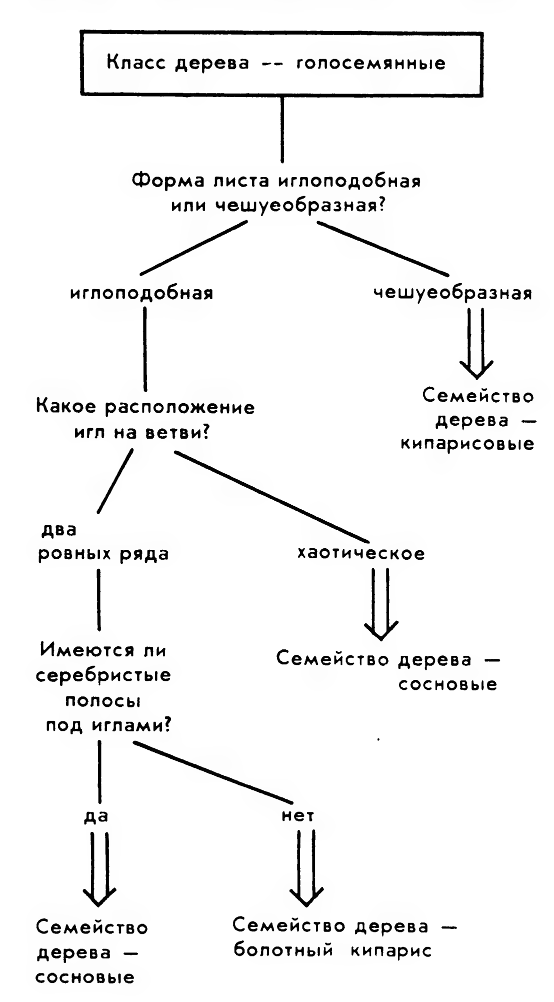


Рис.4. Нахождение семейства неизвестного растения класса голосемянных

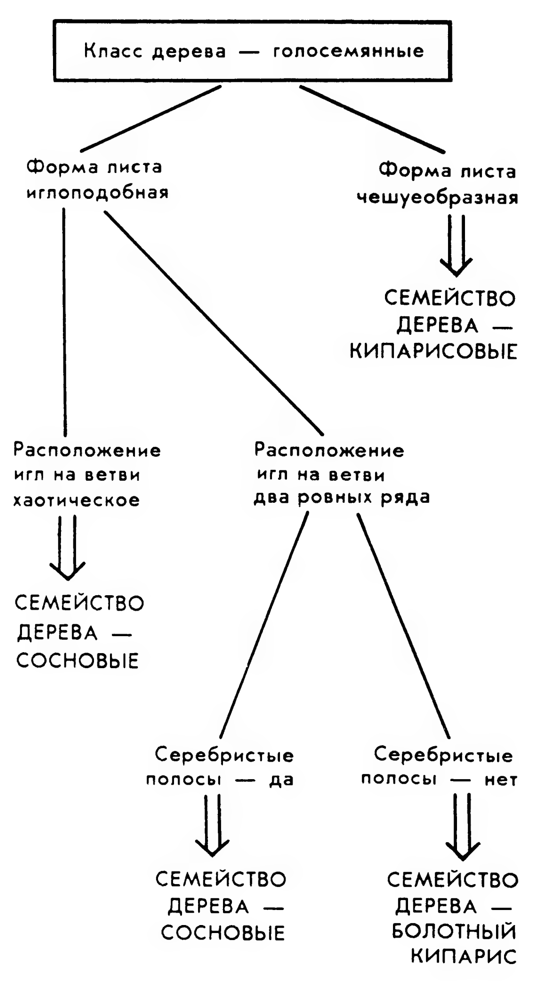


Рис.5. Перевод вопросов и ответов в утверждение фактов

Ее можно использовать для определения растения (травянистое, кустарниковое, лиановое или дерево, а также класса рассматриваемого дерева (голосемянные или покрытосемянные) и конкретного семейства дерева в классе голосемянных (кипарис, болотный кипарис или сосна). На другие вопросы ответы будут получены только после построения остальных ветвей дерева решения.

При разработке конкретного дерева решения попытайтесь разбить задачу на ряд подзадач, для каждой из которых нужно разработать свое дерево решения. Это упростит создание и отладку вашей базы знаний.

Теперь на примере задачи об идентификации растения рассмотрим как дерево решения можно представить в виде правил.

Формирование правил на основе дерева решения. Первым этапом формирования правил является перевод дерева решения из вопросов и ответов в утверждение фактов, что показано на рис.5.

Утверждение фактов состоит из следующих частей:

Название	Пример	Описание
Атрибут	Форма листа	Ключевое слово или фраза, выбранная для представления факта, который мы пытаемся определить
Значение	Иглоподобная	Описание, назначенное атрибуту
Предикат	Является	Элемент, указывающий на отношение между значением и атрибутом

Каждое утверждение типа "форма листа является иглоподобной", называется предложением. Каждая ветвь дерева от ее начала до конца образует правило. Предложения в дереве, расположенные до стрелки образуют условную часть правила, а предложения после стрелки — заключение правила. Ветвь, в которой нет заключения не может быть представлена в виде правила.

На втором этапе формирования правил выполните следующие действия:

- напишите по одному правилу для каждой ветви дерева решения, оканчивающейся заключением;
- присвойте правилу номер;
- начните правило со слова IF (если);
- начиная с левой части дерева, запишите каждое предложение выбранной вами ветви, соединяя их между собой словом AND (и);
- перед заключительной частью правила поставьте слово THEN (то);
- предложения заключительной части соединяйте словом AND (и);
- конец правила — это точка, стоящая в начале строки;

Более подробно о составлении правил рассказано ниже.

Создание базы знаний

Для создания базы знаний можно использовать любой текстовой редактор. Как только вами разработан набор правил, его можно ввести с помощью текстового редактора, который не включает в текстовой файл скрытые символы. Примерами таких редакторов могут быть Turbo Pascal и WordStar. Так, в редакторе WordStar файл необходимо открыть командой "N" (открыть файл, не являющийся документом), которая запрещает включение признака конца страницы и других символов в текстовой файл.

База знаний всегда должна иметь расширение "KB". Это позволяет Микроэксперту отличать файлы баз знаний от других файлов на дискете.

Раздел правил. Рассмотрим простой пример правила, взятый из базы знаний SAMPLE, который поясняет форму представления правил в базе знаний.

9	. 9
IF stem IS	Если стебель
woody	древесный
AND position	И положение —
IS upright	стоит прямо
AND one main	И один основной
trink IS yes	стебель — да
THEN type of	То тип расте-
plant IS tree	ния — дерево.

Правило состоит из следующих элементов:

- Номер правила (служит для идентификации правила) — в нашем примере это 9.

- Условие: это часть правила, содержащая утверждения, которые должны быть оценены. Условие начинается ключевым словом IF, за которым следует ряд предложений, соединенных ключевым словом AND. В нашем примере это

IF stem IS woody
AND position IS upright
AND one main trink IS yes

- Заключение: утверждение в этой части правила выполняется, когда все утверждения в условии верны. Заключение всегда начинается словом THEN и состоит из ряда предложений, объединенных ключевым словом AND. В нашем примере это последняя строка: *ТО тип растения — дерево.*

- Предложение: условие и заключения состоят из ряда предложений, соединенных словом AND. В свою очередь, предложение состоит из атрибута, предиката (отношения) и значения функции. Примеры:

stem IS woody	стебель — древесный
position IS upright	положение — стоит прямо
one main trunk IS yes	один основной стебель — да
type of plant IS tree	тип растения — дерево

- Атрибут: это ключевое слово или фраза, описывающая некоторое качество, о котором мы пытаемся найти информацию. Атрибут представляет собой набор из 20 символов (не более), которые не должны содержать слова !S. В нашем примере это

stem	стебель
position	положение
one main trunk	один основной стебель
type of plant	тип растения

Предикат: устанавливает отношение между атрибутом и его значением. Единственно правильным предикатом в Микроэксперте является IS.

Значение: используется для описания атрибута. Значение представляет собой группу не более чем из 20 символов, которые и не должны содержать слова AND.

"." : точка означает конец правила. Она должна быть первым символом в строке, следующим за последней строкой правила.

Кратко синтаксис правила можно записать следующим образом:

Номер правила (цифра)
IF атрибут IS значение
[AND атрибут IS значение]
THEN атрибут IS значение
[AND атрибут IS значение]
. *«ВК»*

Предложения, заключенные в квадратные скобки, необязательны. В правиле может быть несколько таких предложений. Однако с увеличением количества предложений правило становится менее понятным и отладка его сложнее.

Правило можно вводить, используя любую комбинацию строчных и заглавных букв. Каждое предложение необязательно помещать с новой строки, хотя это делает правила более читабельными. Если после номера правила не было возврата каретки, то между ним и словом IF необходимо вставить пробел.

Включение подсказок в базу знаний. Подсказка — это вопрос, который появляется на экране для получения от пользователей информации о некотором атрибуте, значение которого неизвестно. Подсказка представляет собой строку символов и вводится в базу знаний с помощью текстового редактора. Атрибут дол-

жен иметь только одну подсказку, ассоциируемую с ним. Если нет вопроса об отдельном атрибуте, то он не имеет подсказки. Например, атрибуты TYPE, FAMILY, CLASS не имеют подсказок.

Рассмотрим пример подсказки в базе знаний.

prompt положение

Положение стебля — стоит прямо или стелется ?

В общем виде подсказку можно записать:

[numeric] prompt (атрибут) «ВК»

(строка или строки подсказки) «ВК»

. «ВК»

Элементы в квадратных скобках необязательны.

Если используется слово numeric, то система ожидает от пользователя числовой ответ на подсказку.

Упрощение правил чтения с переводом. Правила, записанные в сжатой форме, могут быть непонятны для профессионального пользователя. Атрибуты, вводимые в виде коротких фраз, позволяют минимизировать объем памяти, необходимый для хранения правила, и ускорить его ввод. Для того чтобы сделать правило более понятным и информативным, каждому атрибуту можно поставить в соответствие перевод. Перевод — это строка символов некоторой длины, которая применяется для пояснения атрибута.

Рассмотрим, как работает перевод.

Предложение — "Стебель является зеленым".

Атрибут — "Стебель".

Возможные значения: "зеленый, древесный".

Перевод для атрибута: "стебель растения".

Перевод предложения: "Стебель растения является зеленым".

В этом примере при появлении в правиле атрибута "стебель" он будет заменен переводом "стебель растения", после чего для окончания предложения система добавит предикат "является" и значение, соответствующее данному атрибуту.

Перевод может быть введен в базу знаний в следующей форме:

trans (атрибут) «ВК»

(строки перевода) «ВК»

. «ВК»

Правила, подсказки и переводы могут быть перемешаны в базе знаний в любом порядке. Каждый атрибут может иметь только один перевод. Когда разрабатывается большая база знаний, проще отладить правила до включения переводов.

Общий вид перевода:

trans (атрибут) «ВК»

(строки перевода) «ВК»

. «ВК»

С использованием переводов становятся более понятными ответы системы на команды HOW, WHY, RULE.

Тестирование и отладка базы знаний

После того как база знаний создана, необходимо проверить, как она работает. В этом разделе показано, как это сделать с помощью программы CROSSREF.

Большую базу знаний надо разрабатывать и отлаживать по частям. Как только введена часть базы знаний, охватывающая законченную проблему, ее необходимо проверить и только после этого вводить следующую. База знаний SAMPLE служит очень хорошим примером такой технологии. Хотя наша цель определить семейства всех деревьев, лиан, кустарников и трав, мы ограничились только семействами хвойных деревьев.

Просмотр базы знаний. Для исправления ошибок нужно получить листинг базы знаний. В операционной системе MS DOS это, например:

TYPE «имя накопителя» : «имя базы знаний» «ВК»

После имени базы знаний обязательно указывается расширение ".KB"

Наиболее распространенные синтаксические ошибки:

- пропущена точка "." в конце правила, подсказки или перевода (помните, что точка должна стоять в первой позиции сроки, следующей

сразу за последней строкой элемента базы знаний);

- отсутствует IF или THEN внутри правила;
- пропущено AND между двумя предложениями;
- если не используете процедуры или функции (см.ниже), то могут быть опущены кавычки в начале или в конце выражения.

Если ошибка обнаружена в процессе работы Микроэксперта, необходимо заметить номер последнего правила, считанного благополучно, и проверить следующее по номеру правило. Если вы полностью исправили эти и другие синтаксические ошибки, то задача отладки базы знаний становится значительно проще.

Использование утилиты CROSSREF. Другим полезным инструментом отладки базы знаний является программа CROSSREF, размещенная на системной дискете. Эта программа может быть использована для получения листинга перекрестных ссылок для некоторой базы знаний. Она создает алфавитный список всех атрибутов параллельно с их переводами, подсказками и всеми значениями, используемыми для каждого из атрибутов. Этот список показывает также, сколько правил используют каждую пару атрибут-значение. С помощью CROSSREF можно обнаружить несовместимость в случае незначительного изменения в имени арибута или его значения.

Для работы с программой CROSSREF необходимо в ответ на подсказку PC DOS набрать CROSSREF. Пример работы с программой представлен на рис. 6.

Обнаружение логических ошибок в базе знаний. Для тщательной проверки базы знаний необходимо иметь листинг базы знаний, листинг перекрестных атрибутов, и если при разработке правил использовалось дерево решений, то и копию этого дерева. При наличии дерева решений очень просто начать консультацию и отвечать на подсказки, следуя по очереди по каждой ветви дерева решений. Если вы аккуратно разрабатывали правила, то вполне возможно окажется, что большинство ошибок — синтаксические.

База знаний (Введите "Quit" для окончания работы с программой): sample <ВК>

ЧТЕНИЕ SAMPLE.KB

Выдать листинг на (<ВК> для терминала) <ВК>

Атрибут: ШИРОКИЕ И ПЛОСКИЕ

ПОДСКАЗКА:

Форма листьев широкая или плоская?

Значение правило(а)

НЕТ 6

ДА 5

Нажмите <Esc> для выхода из программы или другую клавишу для продолжения

Атрибут: КЛАСС

Значение Правило(а)

ПОКРЫТОСЕМЯННЫЕ 5

ГОЛОСЕМЯННЫЕ 1 2 3 4 6

Нажмите <Esc> для выхода из программы или другую клавишу для продолжения

Рис.6. Фрагмент работы с программой CROSSREF.COM (в переводе)

Если правила составлялись без дерева решений или каких-либо других структур, то необходимо набросать план просмотра каждой ветви в ходе консультации. Поскольку целью проверки базы знаний является просмотр всех возможных ее частей, это поможет вам отмечать уже пройденный путь. Бывает полезно перепроверить путь, оказавшийся неудовлетворительным. Отладка базы знаний будет значительно проще, если подходить к этому методично и организованно.

Если консультация прервалась, необходимо сразу изолировать ошибку

путем повтора консультации и ретрассировки тех шагов, которые привели к ошибке.

База знаний (Введите "Quit" для окончания работы с программой): sample «ВК»

Чтение SAMPLE.KB

Выдать листинг на («ВК» для терминала) «ВК»

Атрибут: Широкие и плоские

Подсказка:

Форма листьев широкая и плоская?

Значение правило(а)

НЕТ 6

ДА 5

Нажмите «Esc» для выхода из программы или другую клавишу для продолжения.

Атрибут : Класс

Значение Правило(а)

Покрытосемянные 5

Голосемянные 1 2 3 4 6

Нажмите «Esc» для выхода из программы или другую клавишу для продолжения.

Рассмотрим пример, как с помощью команд Микроэксперта WHY, HOW, WHATIF и RULE (см. приложение) можно обнаружить источник ошибки. Начнем консультацию сразу с места появления ошибки (текст английский):

Форма листа иглоподобная или чешуеобразная? иглоподобная «ВК»

Определены следующие факты:

1) КЛАСС	—	ГОЛОСЕМЯННЫЕ
2) ШИРОКИЕ И ПЛОСКИЕ	—	НЕТ
3) ТИП РАСТЕНИЯ	—	ДЕРЕВО
4) ОДИН ОСНОВНОЙ СТЕБЕЛЬ	—	ДА
5) ПОЛОЖЕНИЕ	—	СТОИТ ПРЯМО
6) СТЕБЕЛЬ	—	ДРЕВЕСНЫЙ

Не найдено значение для атрибута СЕМЕЙСТВО.

Выяснилось, что не может быть найдено ответа в той части базы знаний, которую мы считали законченной.

Do, you want to perform another consultation with this Knowledge Base? Y ("Хотите провести другую консультацию с базой знаний? Да.")

Для обнаружения ошибки начнем консультацию заново и посторно пройдем те шаги, которые привели к ошибке. Как только будет достигнуто правило, выполняемое последним перед концом консультации, необходимо использовать команды Микроэксперта и попытаться локализовать ошибку.

Is the leaf shape needlelike or scalelike? why «ВК»

The system trying to find a value for FAMILY using rule no 1

Rule 1

IF

1) CLASS	IS GYMNOSPERM
2) LEAF SHAPE	IS SCALELIKE THEN
3) FAMILY	IS CYPRESS

Форма листа иглоподобная или чешуеобразная? почему «ВК»

Система пытается отыскать значение для атрибута СЕМЕЙСТВО, используя правило 1

Правило 1

ЕСЛИ

1) Класс	—	голосемянные
2) Форма листа	—	чешуеобразная
ТО		
3) Семейство	—	кипарисовые

С помощью команды WHY находим, что текущая цель — FAMILY, текущее правило — 1. Значение, которое было присвоено LEAF SHAPE до того, как система закончит работу, NEEDLELIKE, т.е. правило 1 является ложным. Теперь необходимо найти следующее правило, которое должно быть выбрано для оценки.

Форма листа иглоподобная или чешуеобразная? что если иглоподобная «ВК»

Возможны следующие заключения:

Правило 2		
1) Семейство	—	сосновые
Правило 3		
2) Семейство	—	сосновые
Правило 4		
3) Семейство	—	сосновые

В ответ на команду "что если NEEDLELIKE" распечатываются прави-

ла, которые могут быть извлечены. Так как из команды WHY известно, что текущая цель — FAMILY, то можно сделать вывод, что следующим будет оцениваться первое правило списка.

Форма листа иглоподобная или чешуеобразная? правило 2 «ВК»
Правило 2
ЕСЛИ

1) класс	— голосемянные
2) форма листа	— иглоподобная
3) расположение игл	— хаотическое
ТО	
4) семейство	— сосновые

Из правила 2 видно, что следующим за LEAF SHAPE (форма листа) будет определяться атрибут NEEDLE PATTERN (расположение игл). Поскольку он не был найден, необходимо определить, как он мог быть доказан.

Форма листа иглоподобная или чешуеобразная? how3«ВК»

Расположение игл — хаотическое не может быть определено с помощью этой базы правил.

Итак, атрибут NEEDLE PATTERN не может быть определен с помощью данной базы знаний. Следует либо добавить правило, обеспечивающее необходимую информацию, либо подсказку, опрашивающую пользователя об этом атрибуте. Отсутствие подсказок — один из частых источников ошибок.

Этот пример показал, как использовать команды WHY, HOW, WHAT IF, RULE и WHAT для выявления ошибок в базе знаний.

Механизм логического вывода

В этом разделе описано применение в Микроэксперте обратного логического вывода для управления консультацией. Представление о том, как работает система, сделает разработку и отладку вашей базы знаний более эффективной.

Карточный механизм логического вывода (МЛВ). Для упрощения понимания того, как используются отдельные части базы знаний в процесссе консультации, воспользуемся моделью МЛВ на основе индексных карточек, каждая из которых представляет определенный элемент базы знаний. Такими элементами могут быть: правила, атрибуты и их значения. Карточка атрибута содержит его имя и подсказку, если она есть. В дальнейшем для простоты из карточек атрибутов будут исключены переводы. Рассмотрим работу карточного МЛВ на примере рис.7.

Стек целей. Первым этапом любой консультации является выбор цели консультации (шаг 1 на рис.7.). В нашем примере конечной целью является FAMILY (семейство). Для начала отложим карточку FAMILY в отдельную "колоду", называемую стеком целей. Атрибут, находящийся в вершине стека, называют текущей целью.

Стек контекстов. В процессе консультаций необходимо запоминать последовательность всех известных фактов. Если атрибут имеет значение, его карточка откладывается в колоду, называемую стеком контекстов. Кроме того, карточку со значением этого атрибута помещают рядом с ним. Данная карточка может содержать правило, в котором атрибуту присваивается его значение либо значение, присвоенное атрибуту пользователем. При извлечении правил из базы знаний их также необходимо удалять из колоды.

Выбор правила для оценки. Прежде всего попытаемся найти карточку с правилом, которое в своей заключительной части (утверждение, следующее за ключевым словом THEN) дает некоторую информацию об атрибуте текущей цели. Процесс поиска решения по заключению называется обратным логическим выводом. Он заключается в том, что система, пытаясь доказать Z, отыскивает сначала факты для доказательства Z. Если правило, содержащее в своем заключении атрибут текущей цели, найдено, то можно перейти к оценке правила. В противном случае необхо-

Шаг	Анализируемое правило	Значение правила	Ответ на подсказку	Стек целей:		Контекстный стек		№ правила	№ отложенного правила
				Признак	№ правила	Признак	Значение		
1				Семейство					
2	1	Н		Класс	1				
3	5	Н		Тип	5				
4	7	Н		Стебель	7				
5	нет		Древесный	Тип	6	Стебель	Древесный		
6	7	Л							7
7	8	Н	Положение		8				
8	нет		Прямой	Тип	5	Положение	Прямой		
9	8	Л							8
10	9	Н		Осн.ствол	9				
11	нет		Да	Тип	5	Осн.ствол	Да		
12	9	И		Класс	1	Тип	Дерева	9	
13	5	Н		Шир. + Плос.	5				
14	нет		Нет	Класс	1	Шир. + плос.	Нет		
15	5	Л							5
16	6	И		Семейство		Класс	Голосем.	6	
17	1	Н		Форма листа	1				
18	нет		Чешуеобр.	Семейство		Форма листа	Кипарис		
19	1	И				Семейство	Чешуеобр.	1	

Рис.7. Пошаговое представление сеанса консультации с помощью МЛВ, иллюстрирующее изменение содержимого стеков (здесь Н — неизвестен, Л — ложь, И — истина)

димо обратиться к пользователю для получения информации.

Оценка правила. В выбранном для оценки правиле все утверждения, стоящие в условной части одно за другим, сравниваются с уже известными фактами. Эти факты находятся в стеке контекстов. В процессе сравнения может возникнуть одна из трех ситуаций:

Случай 1. Значение правила неизвестно. Значение правила является неизвестным, когда один или несколько атрибутов в условии текущего правила отсутствует в стеке контекстов (правило 1, шаг 2 на рис.7). Другими словами, карточки с их значениями не найдены. Карточка с неизвестным атрибутом помещается в список целей вместе с карточкой, содержащей неизвестное правило. Цели, которые в списке целей следуют за конечной целью, иногда называют подцелями. Далее выбирается следующее правило для оценки.

Случай 2. Определение ложного правила. Правило является ложным, если одно из предложений условия противоречит факту в стеке контекстов. (В нашем примере при оценке правила 7 в шаге 6 значение GREEN (зеленый) для атрибута STEM (стебель) противоречит информации о STEM, полученной в шаге 5. В этом случае карточка текущего правила помещается в колоду извлеченных правил. Далее выбирается следующее правило для оценки.

Случай 3. Определение истинного правила Правило считается истинным, если каждое из предложений условия соответствует элементу стека контекстов (правило 9 установлено истинным в шаге 12). Если все предложения в условной части правила истинны, то заключительная часть также должна быть истинна. Так как это правило было выбрано для оценки,

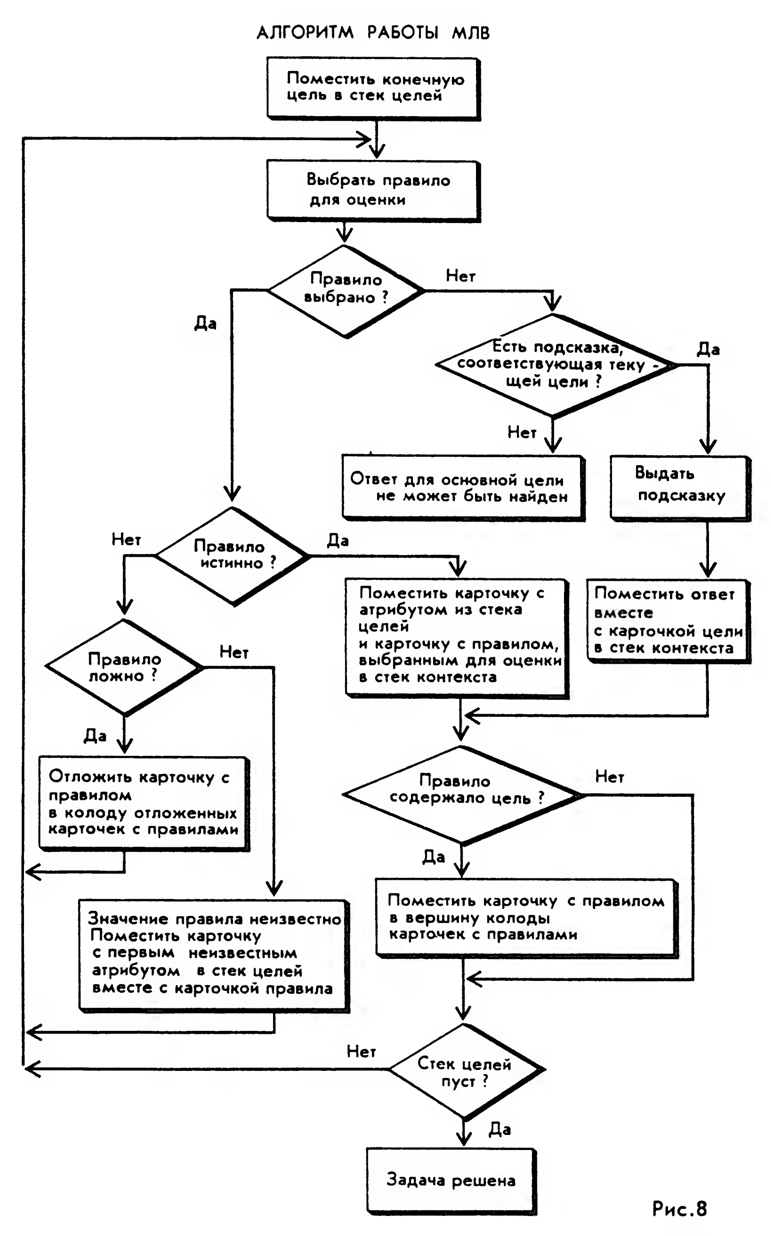


Рис.8

потому что оно содержало текущую цель о заключительной части, удалим эту цель из вершины стека целей и поместим ее в стек контекста. Карточка с правилом, извлеченным ранее из стека целей, должна быть вновь помещена в вершину стека целей и переоценена. Карточка с правилом, которое только что было определено (правило 9), помещается вместе с атрибутом в список контекста для показа значения атрибута и того, как оно было достигнуто. Эта карточка в дальнейшем может быть использована для ответов на вопросы об этой консультации. Если в стеке целей не осталось других целей, то задача консультации считается решенной. В противном случае необходим возврат к выбору правила для оценки.

Опрос пользователя. В случае, когда не найдено правило, содержащее информацию о текущей цели, пользователь должен обеспечить эту

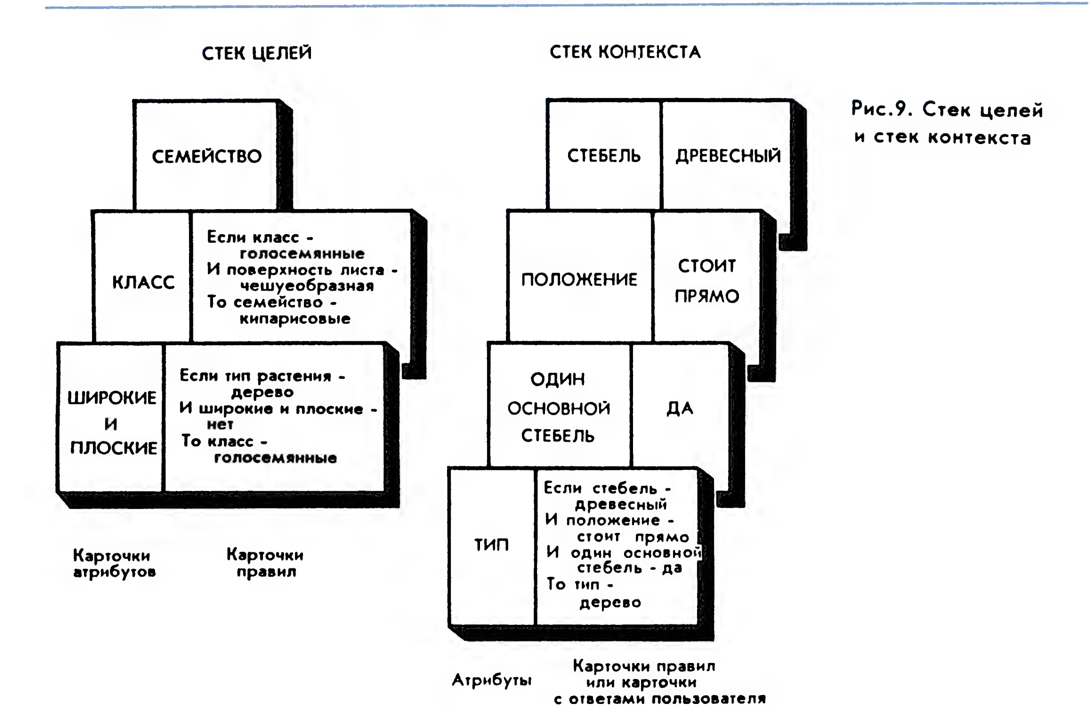


Рис.9. Стек целей и стек контекста

необходимую информацию (в шаге 5 мы опрашиваем пользователя, поскольку отсутствует правило с атрибутом STEM в его заключительной части). Именно здесь следует воспользоваться подсказками, помещенными на карточках атрибутов. Плохо, если на карточке нет подсказки. В этом случае атрибуту не может быть присвоено значение, следовательно, цель консультации не будет достигнута. Это означает, что либо упущена какая-то ветвь дерева решения, либо в правиле есть ошибка.

Когда на текущей карточке цели появилась подсказка, пользователь должен дать ответ. Как только пользователь предоставил недостающую информацию, текущая цель извлекается из стека целей и помещается в стек контекста. Если есть правило с этой целью, оно должно быть помещено на вершину стопки карточек с правилами. Кроме того, необходимо иметь карточку со значением и вместе с карточкой атрибута поместить ее в стек контекста. Как только это все сделано, можно вернуться к выбору правила для оценки. На рис.8 приведен алгоритм работы карточного МЛВ. Если вы сделали карточки, используйте их в процессе работы.

WHY и HOW. Часто пользователю необходимо узнать, почему в ходе консультации был задан определенный вопрос. На рис.9 показаны стек целей и стек контекста так, как они должны выглядеть в шаге 13 нашего примера.

После просмотра всех правил не найдено ни одного, содержащего информацию об атрибуте BROAD AND FLAT (широкий и плоский). Выдаем подсказку о нем. Прежде чем ответить, пользователь может узнать, почему был задан этот вопрос. Для этого необходимо просмотреть не следующую, а предыдущую цель из стека целей. Причиной того, что отыскивается значение атрибута BROAD AND FLAT, является необходимость получения части информации для достижения цели CLASS (класс) в правиле 5. Об этом и сообщает система в ответ на команду WHY. Если теперь пользователь захочет узнать, зачем нужен CLASS, то процесс повторится и из стека целей будет извлечен еще один элемент.

Возвратимся к рассмотренному

выше примеру. Пользователя может также интересовать, каким образом найдено, что тип растения — дерево. Ответом на этот вопрос по команде HOW может быть положение атрибута в стеке контекста. Если рядом с карточкой атрибута есть карточка с правилом, то значение для этого атрибута, помещенное в стек контекста, было взято из этого правила. Для ответа на заданный вопрос мы должны воспользоваться информацией из правила. Если в стеке контекста нет карточки с правилом, то надо сообщить пользователю, что факт был получен из ответа на подсказку.

Введение вопросов WHY и HOW делает консультацию более понятной.

Словарь ключевых слов Микроэксперта

WHY — Эта команда позволяет просмотреть текущую цель и правило, с помощью которого отыскивается эта цель. Каждое повторное использование команды выявляет предыдущую цель и используемое правило, пока не будет достигнута главная цель.

HOW(n) — Строки, распечатываемые каждой командой Микроэксперта, пронумерованы. Когда вводится эта команда и номер строки, система распечатывает, как было доказано, данное предложение, или если оно неизвестно, как это предложение может быть доказано. Предложение может быть доказано либо с помощью правила базы знаний, либо на основании информации, полученной от пользователя.

WHATIF (ответ) — Эта команда позволяет увидеть, какие заключения может сделать система на основании указанного ответа на текущую подсказку.

RULE(n) — Команда распечатывает правило с указанным номером.

WHAT — Команда распечатывает все уже выявленные в ходе консультации факты.

QUIT — Эта команда используется для завершения текущей консультации.

Литература

1. Thompson B.A., Thompson W.A. Micro Expert, Version 1.0, IBM PC. McGray-Hill Boor Company, New-York, StLouis, San-Francisco, Hamburg, 1985.

2. Реальность и прогнозы искусственного интеллекта: Сб. статей;/Пер. с англ. Под ред. и с предисловием В.Л.Стефанюка. — М.: Мир, 1987.

3. Уотерман Д. Руководство по экспертным системам/Пер.с англ. Под ред. В.Л.Стефанюка. — М.: Мир, 1989.

4. Сойер Б., Фостер Д.Л. Программирование экспертных систем на Паскале: Пер.с англ.; предисловие В.П.Иванникова. — М.: Финансы и статистика, 1990.



Ласкин С.А.
(г.Жуковский)

МОНТЕ-КАРЛО

Описание игры

В игре может участвовать один человек. Играющий делает ставку на некоторое число от 1 до 9, после чего ЭВМ случайным образом подбирает колонку из пяти цифр. Чем чаще выпадает в этой колонке задуманное вами число, тем больше сумма выигрыша. Первоначально вы имеете 3000 руб., ЭВМ — 5000 руб.

Краткое описание программы

Игра основана на функции случайных чисел. Строки 5-20 окрашивают служебную строку в цвет фона для лучшего восприятия игры. В строках 140-150 расположены операторы, рисующие заставку. Строки 170-280 печатают краткое описание игры. Строки 290-600 делают графические построения. Строка 610 делает запрос вашей ставки. Строка 670 анализирует ввод числа ставки: если ставка больше суммы вашего банка, она не засчитывается и печатается шутливая надпись. Строка 675 печатает введенное вами число (ставка). Строки 700-730 делают запрос загадываемого числа. Строка 730 — эхо загадываемого числа. Строки 760-870 — присвоение переменных A, N, C, D, E случайного числа в пределах от 1 до 9. В строках 890-1050 расположены операторы анализа выигрыша. Строки 1050-1060 — суммы банков после игры. Строки 110-1120 — подпрограмма подачи звука. Строки 1140-1150-1210 — вывод шутливых надписей. Строка 1250 — запрос: хочет ли игрок продолжить игру? Строки 1280-1300 — анализ введенной команды: если игрок нажал 0 — игра прекращается, иначе — продолжается.

```
5 COLOR 0%, 0%
10 ? CHR$(158%)
15 DIM A%(4%)
20 COLOR 2%, 4%
30 CLS
40 FOR I% = 1% TO 120%
50 X% = INT(RND(1)*250%)
60 Y% = INT(RND(1)*230%)
70 X1% = INT(RND(1)*250%)
80 Y1% = INT(RND(1)*230%)
90 C% = INT(RND(1)*3%) + 1%
100 LINE (X,Y)-(X1,Y1),C
110 NEXT I%
120 ? AT(10%,11%) "MONTE-KARLO"
    AT(13%,9%) "ИГРА-"
130 ? AT(3%,21%) "Автор:С.Ласкин"
140 X$ = INKEY$
150 IF X$ = " " THEN 140
160 CLS
170 ? AT(7%,9%) "Для информации"
180 ? AT(10%,10%) "Нажмите-" AT(12%,11%)
    "<?>"
190 X$ = INKEY$
200 IF X$ = "?" OR X$ = "/" THEN 220
210 IF X$ = " " THEN 190 ELSE 290
220 CLS
230 ? " ПРАВИЛА ИГРЫ:"
240 ? "Играющий делает ставку на некоторое
    число (от 1 до 9), после чего машина слу-
    чайным образом подбирает колонку из
    пяти цифр. Чем больше в этой колонке
    будет повторяться задуманное Вами"
    250 ?"число, тем больше сумма выигрыша.
    Ваш банк равен 3000.
    Банк БК равен 5000."
260 ? AT(5%,15%) "Желаю удачи!"
    AT(10%,22%) "Нажмите любую клавишу"
270 X$ = INKEY$
280 IF X$ = " " THEN 270
290 CLS
300 I1% = 230%
310 Y% = 250%
320 FOR I% = 0% TO 6%
330 I1% = I1%-1%
340 Y% = Y%-1%
350 LINE (I%,I%)-(Y%,I1%),3%,B
360 NEXT I%
370 LINE (20%,20%)-(230%,70%),1%,B
380 B% = 5000%
390 BEEP
400 LINE (20%,180%)-(230%,210%),1%,B
410 Y% = 100%
420 X% = 190%
430 X1% = 130%
440 LINE (70%,140%)-(180%,170%),3%,B
450 FOR I% = 60% TO 65%
460 Y% = Y% + 1%
470 X% = X%-1%
480 X1% = X1%-1%
490 LINE (I%,Y%)-(X%,X1%),1%,B
500 NEXT I%
510 LINE (80%,75%)-(170%,95%),2%,B
520 P% = 3000%
530 CIRCLE (30%,125%),18%,1%,0,0,1.3
540 PAINT (30%,116%),3%,1%
550 CIRCLE (30%,119%),10%,2%,0.3,4.7,0.7
560 CIRCLE (30%,131%),10%,2%,3.5,1.7,0.7
570 LINE (28%,110%)-(28%,140%),2%
580 LINE (32%,110%)-(32%,140%),2%
590 ? AT(2%,9%) "Monte"
600 ? AT(2%,15%) "Karlo"
610 ? AT(3%,3%) "Ваш банк:" AT(3%,5%) P%
    AT(17%,3%) "Банк БК:" AT(16%,5%) B%
    AT(3%,19%) "              "
    AT(3%,19%) "Ваша ставка: "
620 LOCATE 15%,19%
630 INPUT S%
640 ? AT(17%,16%) "          "
660 BEEP
670 IF S% > P% OR S% < 1% THEN ?
    AT(3%,19%) "ЖУЛИК!"
    ELSE GOSUB 1100
675 ? AT(9%,15%) "Ваша" AT(9%,16%)
    "СТАВКА:" S%
680 BEEP
690 IF S% > P% OR S% < 1% THEN 610
700 ? AT(3%,19%) "Загадывайте число:"
710 LOCATE 21%,19%
720 INPUT X%
730 ? AT(11%,8%) "Число:" X%
740 ? AT(3%,19%) "
750 COLOR 3%,4%
755 W% = 0%
```

```
760 FOR I% = 1% TO 15%
770 FOR J% = 0% TO 4%
780 A%(J%) = INT(RND(1)*9%) + 1%
790 ? AT(10% + J% + J%,11%) A%(J%)
800 IF I% = 15% AND A%(J%) = X% THEN
    W% = W% + 1%
810 NEXT J%,I%
880 COLOR 2%,4%
```

КУПЮРА

```
950 IF W%<2% THEN V% = -S% ELSE
    V% = (W%-2%)*500% + S%
960 B% = B%-V%
970 P% = P% + V%
1050 IF W%<2% THEN ? AT(3%,19%) "Не
    повезло!"
    ELSE ? AT(3%,19%) "Вам повезло!"
1060 ? AT(3%,5%) P% "   " AT(16%,5%)
    B% "   "
1080 GOSUB 1100
1090 GOTO 1140
1100 ? STRING$(40%,7%)
1130 RETURN
1140 IF W%<2% THEN ? AT(3%,19%) "ВАМ
    НЕ ВЕЗЕТ!"
1145 GOSUB 1100
1150 IF B%< =0% THEN ? AT(3%,19%) "Я
    РАЗОРЕН!"
    ELSE 1210
1160 GOSUB 1100
1170 GOTO 1210
1180 ? AT(3%,19%) "НАЖМИТЕ ЛЮБУЮ
    КЛАВИШУ"
1190 X$ = INKEY$
1200 IF X$ = " " THEN 1190 ELSE 160
1210 IF P%< =0% THEN ? AT(3%,19%) "УВЫ,
    ВЫ РАЗОРЕНЫ!"
    ELSE 1250
1220 GOSUB 1100
1230 GOTO 1180
1250 ? AT(3%,19%) "ПРОДОЛЖИМ? "
1260 GOSUB 1100
1270 ? AT(3%,19%) "(0 — НЕТ, НЕ 0 — ДА)"
1280 X$ = INKEY$
1290 IF X$ = " " THEN 1280
1300 IF X$ = "0" THEN 30 ELSE 610
1310 END
```

Заключение рецензента

Программа, предложенная С. Ласкиным, работоспособна, но ее можно ускорить и несколько сократить по объему. Для этого следует заменить все переменные на целочисленные. Вместо переменных A, N, C, D, E можно ввести массив A%(4%). Это позволяет существенно сократить участок программы 760-1050.

Звуковой сигнал в подпрограмме, начинающейся со строки 1100, можно подать одним оператором

1100 ? STRINGS$ (40%,7%)

Текст модифицированной программы предложен здесь читателям. Она работает быстрее.

Теперь можно оценить, имеется ли у игрока выигрышная стратегия игры. Для этого сначала определим вероятности каждого из шести интересующих нас событий. Выпадения всех девяти цифр будем считать равновероятными. Эти вероятности равны:

$Po = (8/9)^5 \sim 0,5549$ вероятность непопадания в строку задуманной цифры;

$P1 = (1/9)(8/9)^4 C^1{}_5 \sim 0,3468$ — вероятность попадания цифры в строку только один раз;

$P2 = (1/9)^2 (8/9)^3\, C^2{}_5 \sim 0,0867$ — два раза;

$P3 = (1/9)^3 (8/9)^2 C^3{}_5 \sim 0,0108$ — три раза;

$P4 = (1/9)^4 (8/9) C^4{}_5 \sim 0,0007$ — четыре раза;

$P5 = (1/9)^5 \sim 0,000017$ — пять раз.

Как нетрудно видеть, сумма вероятностей полной группы событий равна единице. Зная вычисленные вероятности, можно рассчитать вероятность проигрыша ставки и математическое ожидание величины выигрыша.

Вероятность проигрыша ставки равна $P_n = P_o + P_1 = 0,9017$

Эта вероятность, точнее ее величина, говорит о достаточно длительном ожидании игроком счастливого события: в среднем выигрыш ждет игрока один раз из десяти игр (ставок).

Математическое ожидание величины выигрыша игрока в одной попытке равно (S — размер ставки)

$$S = 0,9017\ (-S) + 0,0867 \cdot S + 0,0108\ (S + 500) + 0,0007\ (S + 1000) + 0,000017\ (S + 1500) = (-0,9017 + 0,0867 + 0,0108 + 0,0007 + 0,000017)\ S + 0,0108 \cdot 500 + 0,0007 \cdot 1000 + 0,000017 \cdot 1500 \sim 6,1255 - 0,8035 \cdot S$$

Таким образом, чем меньше средняя ставка, тем больше средний выигрыш игрока. Граничное значение ставки, при которой игрок будет разоряться так же часто, как и разорять БК при достаточно длительной игре, можно приблизительно определить из уравнения

$$6,1255 - 0,8035\ S_{гр} = 0\ ,$$

$$S_{гр} = 7,62.$$

Следовательно, выигрышная стратегия игрока достаточно проста: нужно все время делать ставки равные 1 — 7. При этом не будем мелочиться и ограничим размер ставок целыми значениями.

Если знать эту стратегию, то привлекательность игры существенно снижается. Так как при такой стратегии выигрыш игрока практически гарантирован, то играть с этим "одноруким бандитом", предложенным С.Ласкиным, на деньги можно.

Более точный анализ игры связан с применением теории блуждания случайной точки и при органичении банков БК и игрока не является тривиальной задачей.

Л. Н. Жариков



Э 41 **Экспертные системы**. — М.: Знание, 1990. — 48 с. — (Новое в жизни, науке, технике. Сер. "Вычислительная техника и ее применение"; № 10).
ISBN 5-07-000529-4
20 к.

В брошюре рассказано об одной из основных составляющих искусственного интеллекта — экспертных системах. Изложены история вопроса, современное состояние и перспективы развития.
Материал рассчитан на широкий круг читателей.

2404000000 **ББК 32.97**



Научно-популярное издание

ЭКСПЕРТНЫЕ СИСТЕМЫ

Гл.отраслевой редактор *Г. Г. Карвовский*
Редактор *Б. М. Васильев*
Мл.редактор *Н. А. Васильева*
Художник *В. Н. Конюхов*
Худож.редактор *И. А. Емельянова*
Техн.редактор *Т. В. Луговская*
Корректор *В. И. Гуляева*

ИБ № 10567

Подписано к печати 2.10.90. Формат бумаги 70x100 1/16. Бумага офсетная. Печать офсетная. Усл.печ.л.3,90. Усл.кр.-отт.8,45. Уч.-изд.л.4,20. Тираж 71575 экз. Заказ 1601. Цена 20 коп. Издательство "Знание". 101835, ГСП, Москва, Центр, проезд Серова, д.4. Индекс заказа 904710. Отпечатано с оригинал-макета издательства "Знание" на ордена Трудового Красного Знамени Тверском полиграфическом комбинате Государственного комитета СССР по печати. 170024, г.Тверь, пр.Ленина,5.

Цена 20 коп.

Индекс 70195

Адрес подписчика:

(10-32)



Подписная научно-популярная серия
ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА
И ЕЁ ПРИМЕНЕНИЕ

"Для одних экспертные системы представляют "светлую надежду" информационной технологии, для других — больше напоминают современный вариант "желтой опасности", поскольку японцы могут сильно обогнать европейцев. Для остальных же — это просто очередной трюк, старое вино в новых бутылках".

Р. Форсайт

Наш адрес:
СССР,
Москва,
Центр,
проезд
Серова, 4